



University of **HUDDERSFIELD**

University of Huddersfield Repository

Muhamedsalih, Hussam

Investigation of Wavelength Scanning Interferometry for Embedded Metrology

Original Citation

Muhamedsalih, Hussam (2013) Investigation of Wavelength Scanning Interferometry for Embedded Metrology. Doctoral thesis, University of Huddersfield.

This version is available at <http://eprints.hud.ac.uk/id/eprint/18090/>

The University Repository is a digital collection of the research output of the University, available on Open Access. Copyright and Moral Rights for the items on this site are retained by the individual author and/or other copyright owners. Users may access full items free of charge; copies of full text items generally can be reproduced, displayed or performed and given to third parties in any format or medium for personal research or study, educational or not-for-profit purposes without prior permission or charge, provided:

- The authors, title and full bibliographic details is credited in any copy;
- A hyperlink and/or URL is included for the original metadata page; and
- The content is not changed in any way.

For more information, including our policy and submission procedure, please contact the Repository Team at: E.mailbox@hud.ac.uk.

<http://eprints.hud.ac.uk/>

INVESTIGATION OF WAVELENGTH SCANNING INTERFEROMETRY FOR EMBEDDED METROLOGY

HUSSAM MAKI MUHAMEDSALIH

A thesis submitted to The University of Huddersfield
in partial fulfilment of the requirements for
the degree of Doctor of Philosophy

The University of Huddersfield

May 2013

Copyright Statement

- i. The author of this thesis (including any appendices and/or schedules to this thesis) owns any copyright in it (the “Copyright”) and s/he has given The University of Huddersfield the right to use such Copyright for any administrative, promotional, educational and/or teaching purposes.
- ii. Copies of this thesis, either in full or in extracts, may be made only in accordance with the regulations of the University Library. Details of these regulations may be obtained from the Librarian. This page must form part of any such copies made.
- iii. The ownership of any patents, designs, trademarks and any and all other intellectual property rights except for the Copyright (the “Intellectual Property Rights”) and any reproductions of copyright works, for example graphs and tables (“Reproductions”), which may be described in this thesis, may not be owned by the author and may be owned by third parties. Such Intellectual Property Rights and Reproductions cannot and must not be made available for use without the prior written permission of the owner(s) of the relevant Intellectual Property Rights and/or Reproductions.

Acknowledgment

All thanks and praises are due to God the Almighty for his blessing that made this work possible and completed.

I would like to express my sincere thanks to my supervisor professor Xiangqian Jiang for firstly providing the opportunity and facilities to undertake this research, secondly for her continued guidance and encouragement throughout this project. Her valuable suggestions during all the phases of this work were indispensable.

Special thanks must also go to Dr. Feng Gao whose enthusiasm and technical guidance have been greatly appreciated, and to Dr. Haydn Martin for his interest and advice throughout this project.

I would also like to thank all my colleagues in the EPSRC Centre for Innovative and Manufacturing in Advanced Metrology, especially Professor Liam Blunt. You all have contributed to creating a unique working environment.

I would also like to express my gratitude to University of Huddersfield for awarding me the fee waiver scholarship to continue my PhD, and to EPSRC Centre for Innovative and Manufacturing in Advanced Metrology for support for the research.

Last, by no means least, my special thanks to Mum, Dad and brothers Sinan, Yousif and Haitham. Without your endless support and unlimited patience, I could have never completed this work.

This work is dedicated to my parents for enduring my absence while I was working on this project.

Abstract

The tremendous growth in the manufacture of a wide range of deterministic and complex free form surfaces, has made surface metrology an essential part of the manufacturing process. The mass production of these surfaces at the micro and nano-scales is stimulating the development of new metrology instrumentation. Current instrumentation still suffers from one fundamental limitation, namely the inability to perform embedded (on-line) measurement in the manufacturing environment. The capability to measure surfaces on-line can improve yields through better process control, thus reducing costs and enabling the commercialisation of more hi-tech products. Better embedded metrology can also enable the manufacture of next generation surfaces with high precisions and more complex artefacts. Embedded measurement requires the development of precise on-line measurement instruments capable of a large measurement range with immunity from environmental vibrational disturbance.

This thesis introduces a new optical interferometry system for fast areal surface measurement at the micro and nano-scales which is robust against the effects of vibration. Wavelength Scanning Interferometry (WSI) together with an acousto-optic tuneable filtering technique has been developed which can measure surfaces with large step heights. The illumination source of the WSI tunes the wavelength provided from a broadband light (halogen-tungsten lamp) over a 93 nm range with a linewidth of approximately 2 nm. This light source is combined with a Linnik interferometer to generate an apparatus that can measure large discontinuous step heights absolutely and without any 2π phase ambiguity and with nano scale resolution.

A reference interferometer which acts as a position feedback sensor is multiplexed with the WSI in order to stabilise the system against environmental disturbances. The reference interferometer is integrated with an active closed loop control system to track the optical path length mechanically using a piezoelectric translator. The stabilised WSI can effectively perform surface measurements under conditions of vibration that would make measurement impossible for an unstabilised interferometer.

High speed measurement is of high importance for embedded metrology instrumentation as this has a direct effect on manufacturing throughput. In order to increase the speed of measurement calculation, this thesis presents a parallel programming model using large numbers of threads processed by many cores of a graphic processing unit (GPU). This model can accelerate the computing time up to 27 times compared to a conventional single threaded CPU calculation. This parallel programming model is based on CUDATM (Compute Unified Device Architecture) program structure developed by NVIDIA and harnesses the cheap computing power of commercial graphics cards. The system design, implementation, operation, fringe analysis algorithms and parallel programming model are all reported in this thesis.

The results of measurements for standard step height samples and a V-groove structured surface are also presented. For the standard step height specimens the system achieved large measurement ranges up to 96 μm with nanometre measurement uncertainty. The measurement results of a step height sample, under mechanical disturbance, showed that the system can successfully withstand environmental vibration up to 300 Hz. For the V-groove structure, the system successfully measured the sample by rotating the interferometer to face the groove side walls using an accurate rotary stage. The measurement result shows that the system has the potential to be used for on-line/in-process measurement on a shop floor.

List of Contents

Abstract.....	4
List of Contents	5
List of Figures.....	8
List of Tables	10
List of Abbreviations	11
List of Nomenclature	12
1. Introduction.....	14
1.1 Overview.....	14
1.2 Aim.....	16
1.3 Objectives.....	16
1.4 Contribution	17
1.5 Thesis Organization	17
1.6 Publications.....	19
2. Surface Metrology.....	20
2.1 Introduction.....	20
2.2 Surface Metrology.....	20
2.3 Early Development of Instrumentation.....	22
2.4 Filtration.....	22
2.5 Characterisation	23
2.6 Development of Modern Instrumentation.....	24
2.6.1 Contact Method.....	24
2.6.1.1 Stylus Based Methods.....	24
2.6.1.2 Scanning Probe Microscopes	25
2.6.2 Non-contact Methods.....	27
2.6.2.1 Interferometry	27
2.6.2.3 Light Scattering.....	30
2.6.2.4 Confocal Microscopy	31
2.7 Comparison Between Instruments	32
2.8 Summary	34
3. Interferometry.....	35
3.1 Introduction.....	35
3.2 Light Interference and Coherence.....	35
3.3 Interferometry for Surface Metrology.....	38
3.3.1 Interferometric objective configurations.....	39
3.4 Phase Shifting Interferometry	41
3.5 Heterodyne Interferometry.....	42
3.6 White Light Interferometry	43
3.7 Two-Wavelength Interferometry	45
3.8 Wavelength Scanning Interferometer	47
3.9 Summary	52
4. Development of WSI instrument	54
4.1 Introduction.....	54

4.2 System Scheme	55
4.3 Development Methodology.....	57
4.3.1 Development of WSI	57
4.3.2 Acceleration the computing process	58
4.3.3 Stabilisation of the interferometry	58
4.4 Summary	58
5. Wavelength Scanning Interferometry: Implementation and Operation	59
5.1 Introduction.....	59
5.2 The WSI Overview	59
5.3 Light Source Setup.....	60
5.3.1 Wavelength Scanning Characterisation	63
5.4 WSI Experimental Setup.....	67
5.5 The WSI Operation and Data Capturing	68
5.6 Intensity Behaviour of AOTF Filtered light	71
5.7 Coherence length of WSI.....	72
5.8 Lateral Range and Resolution	75
5.9 Summary	78
6. Fringe Analysis and Algorithm.....	80
6.1 Introduction.....	80
6.2 Interference Pattern of the Fringe	80
6.3 Analysis of Phase Determining Algorithms.....	83
6.3.1 Phase Shift Determination Using Simple FFT: Algorithm A	84
6.3.2 Phase Shift determination Using Fitted FFT: Algorithm B	85
6.3.3 Interference Pattern Analysis Using Fourier Transform: Algorithm C	86
6.3.4 Localised Peaks of Interference Pattern Using Convolution: Algorithm D.....	93
6.4 Experiment Results and Discussion	97
6.4.1 Measuring Sample Step Height	97
6.4.2 Measuring Sample V Groove.....	105
6.5 Measurement Repeatability.....	110
6.6 Summary	112
7. Accelerating Computing Process Using GPU.....	114
7.1 Introduction.....	114
7.2 GPUs versus CPUs.....	114
7.3 Compute Unified Device Architecture	116
7.3.1 Thread hierarchy	117
7.3.2 Memory hierarchy.....	120
7.4 Acceleration of Computing Process.....	121
7.5 The Proposed CUDA Structure.....	122
7.5.1 Data Arrangement	124
7.5.2 Data Transfer	125
7.5.3 Thread Organisation.....	126
7.5.4 Kernel Invocation and Execution.....	128
7.6 Surface Representation	130
7.7 Result and Discussion	131
7.8 Summary	133
8. Compensation for Environmental Disturbances.....	134
8.1 Introduction.....	134

8.2 System Implementation.....	134
8.3 Block Diagram of the Reference Interferometer.....	137
8.3.1 Controller representation	138
8.3.2 PZT transfer function	139
8.3.3 Interferometer Block Diagram	147
8.4 Simulation of Close Loop Compensated System.....	149
8.5 Stabilisation Practical Performance and Results.....	152
8.6 System Repeatability.....	157
8.7 Summary	159
9. Discussion, Conclusions and Future Work.....	160
9.1 Overview	160
9.2 System performance.....	162
9.3 Major Contribution	163
9.4 Conclusions.....	165
9.5 Future Work	167
9.5.1 Improvement of accuracy and measurement range.....	167
9.5.2 Accelerate the capturing time	167
9.5.3 Accelerate the computing time	168
9.5.4 Increase the stabilisation bandwidth	168
10. References	169
11. Appendix A	175
A.1 AOTF Driving Process.....	175
A.2 WSI Prototype	178
A.2.1 Light Source	178
A.2.2 Interferometer.....	184
A.3 Specifications	188
12. Appendix B	189
B.1 Programs of the four Algorithms: Matlab Code.....	189
B.1.1 Using Simple FFT	189
B.1.2 Using Fitted FFT	190
B.1.3 Fringe Analysis using FFT	191
B.1.4 Localise Peaks Using Convolution	193
B.2 Localise Peaks	194
B.3 Measurement of 9.758 μ m Wyko Sample.....	195
B.4 Measurement of Sine Wave Sample	197
13. Appendix C	199
Appendix C.1 Data Arrangement.....	199
Appendix C.2 Memory Management and Transfer.....	199
Appendix C.3 Thread Organisation	199
Appendix C.4 Main Program Body (Kernel Invocation and Execution).....	199
Appendix C.5 Kernel of Removing the background intensity	200
Appendix C.6 Kernel of Data Filtration.....	200
Appendix C.7 Kernel of Determining the Phase Shift.....	200
Appendix C.8 Kernel of Implementing the Phase Offset Distribution	201
Appendix C.9 Kernel of Correcting the Phase Distribution	201
Appendix C.10 Kernel of Fitting the corrected phase	201
Appendix C.10 Kernel of Calculating the Surface Height.....	202

Appendix C.11 Write the Surface Height Data into SDF file.....	202
Appendix C.12 The main fringe analysis by FFT using CUDA program	203
14. Appendix D	211
D.1 Implementation of PI Controller	211
D.2 Signal Detection and Manipulation.....	212
D.3 PI Controller Circuit.....	213
D.3.1 Subtracting amplifier.....	213
D.3.2 PI controller.....	214
D.4 Visibility Measurement.....	216
D.5 PI Enable	218
D.6 PI Disable	219
15. Appendix E	220
Publications and Awards.....	229

List of Figures

Figure 2.1 Surface measurement stages	21
Figure 2.2 Some numerical parameters to define surface profile	23
Figure 2.3 Principle of stylus instrument	25
Figure 2.4 A sinusoidal surface with a stylus [adapted from Elson and Bennett (1979)]	25
Figure 2.5 Principle of interaction between tip and sample.....	26
Figure 2.6 Multiplexed fibre interferometer [adapted from Dejiro, et al., 2004]	29
Figure 2.7 Laser triangulation method[adapted from (Groover, 2007)]	30
Figure 2.8 Principle of light scattering method.....	31
Figure 2.9 Schematic of confocal microscopy.....	31
Figure 2.10 Amplitude-wavelength plot [adapted from Jiang et al., 2007b]	33
Figure 3.1 Interferometric objective configurations	40
Figure 3.2 Interferogram for scanning white light interferometry	43
Figure 3.3 Schematic of CCI 6000 [adapted from (Kaplonek and Lukianowicz, 2012)]	44
Figure 3.4 Fringes of two wavelength interferometer (a) fringes of $\lambda_a=682.42$ nm (b) fringes of $\lambda_b=590.98$ nm (c) beating signal at λ_s	46
Figure 3.5 Schematic of unbalanced WSI.....	48
Figure 3.6 Simulated interference patterns	49
Figure 4.1 Example of measurement application (a) Master drum [adapted from (UPS2, 2011)] (b) WSI on the machine	54
Figure 4.2 Schematic of the WSI interferometry system.....	56
Figure 4.3 Development strategy	57
Figure 5.1 Basic configuration of the proposed WSI; where DAQ: data acquisition card and PBC: Pulse to binary converter.....	60
Figure 5.2 WSI light source setup.....	61
Figure 5.3 Divergence angle of the collimated white light.....	62
Figure 5.4 Wavelength scanning range versus driving frequency	64
Figure 5.5 Wavelength bandwidth (a) linewidth at 683.42 nm = 2.2 nm, (b) linewidth at 637.39 nm = 2 nm, and (c) linewidth at 590.98 nm = 1.7nm.....	66
Figure 5.6 The WSI experimental setup, where PH: Pinhole, CL: Collimating lens, OF: Optical fibre, COL: Coupling lens, B.S: Beam splitter, OL: Objective lens, RF: Reference mirror, FG: Frame grabber, PBC: Pulse to binary converter	68

Figure 5.7 WSI system.....	69
Figure 5.8 Captured frames by the CCD.....	70
Figure 5.9 Background intensity variation during wavelength scanning.....	72
Figure 5.10 Coherence length of WSI at (a) $\lambda=683.42$ nm (b) $\lambda=637.39$ nm (c) $\lambda=590.98$ nm	74
Figure 5.11 Magnified scale for coherence length of WSI at $\lambda=590.98$ nm.....	75
Figure 5.12 Captured frame for micrometer stage with 50 μ m divisions using a 2X lens	76
Figure 5.13 Relationship between micrometer stage and pixelation for 2X lens	76
Figure 6.1 Interference patterns of two points on the sample	82
Figure 6.2 Simulated Fourier power spectral densities for different heights.....	84
Figure 6.3 Simulated Fourier power spectral density for 7490 nm single point height.....	86
Figure 6.4 Compensated interference pattern	87
Figure 6.5 The Fourier transform spectrum of Equation 6.11	89
Figure 6.6 The Fourier transform spectrum of equation 6.13	89
Figure 6.7 The discontinuous phase distribution as determined	91
Figure 6.8 The offset phase distribution	92
Figure 6.9 The corrected continuous phase distribution	92
Figure 6.10 The fitted continuous phase distribution	92
Figure 6.11 Convolution process	95
Figure 6.12 Convolution output	96
Figure 6.13 Measured results for 4.707 μ m standard step height sample using algorithm A (a) areal measurement (b) cross-sectional profile	100
Figure 6.14 Measured results for 4.707 μ m standard step height sample using algorithm B (a) areal measurement (b) cross-sectional profile	101
Figure 6.15 Measured results for 4.707 μ m standard step height sample using algorithm C (a) areal measurement (b) cross-sectional profile	102
Figure 6.16 Measured results for 4.707 μ m standard step height sample using algorithm D (a) areal measurement (b) cross-sectional profile	103
Figure 6.17 Measured result for 100nm standard step sample using algorithm B (a) areal measurement (b) cross-sectional profile	104
Figure 6.18 Measured result for 100nm step height standard sample using algorithm C (a) areal measurement (b) cross-sectional profile	104
Figure 6.19 Measured result for 100nm step height standard sample using algorithm D (a) areal measurement (b) cross-sectional profile	105
Figure 6.20 V groove sample.....	107
Figure 6.21 The WSI interferometer rotated by 45° to measure V groove sample.....	107
Figure 6.22 Captured frame showing 5 steps return of V groove sample	107
Figure 6.23 Measured results for V groove sample (a) areal measurement (b) cross-sectional profile suffers direction ambiguity	108
Figure 6.24 Mapping for step height direction	109
Figure 6.25 Corrected cross-sectional profile	109
Figure 6.26 Measurement repeatability	111
Figure 6.27 Flow chart of fringe analysis using FFT (algorithm C).....	113
Figure 7.1 Floating point operation per second[adapted from (NVIDIA, 2011)].....	115
Figure 7.2 General CUDA program structure.....	117
Figure 7.3 Thread hierarchy.....	118
Figure 7.4 CUDA code sample of matrices addition (NVIDIA, 2011)	119
Figure 7.5 Relationships between the memory spaces in device memory and threads	120
Figure 7.6 The proposed program structure using CUDA.....	123

Figure 7.7 The frames order is as obtained from the capturing process. n is the number of captured frames, h is the frame height (e.g. 480) and w is the frame width (e.g. 640)	124
Figure 7.8 Data structure of DataHost	125
Figure 7.9 Thread organisation	127
Figure 8.1 Configuration of the proposed compensated interferometer	135
Figure 8.2 Block diagram of the closed loop system. The GP(s), Gc(s) and GPZT(s) are transfer functions of interferometer, controller and the PZT respectively	137
Figure 8.3 Block diagram of the controller	139
Figure 8.4 Simple PZT and mass model (b) directions of forces affects a PZT element (c) PZT model response in direction 3 only	139
Figure 8.5 Mechanical spring-damper system	142
Figure 8.6 Open loop step response experimental setup	144
Figure 8.7 Operating region of piezo step response	145
Figure 8.8 Step response of the PZT	145
Figure 8.9 Block diagram of the PZT	146
Figure 8.10 Block diagram of the interferometer	147
Figure 8.11 Fringe visibility	148
Figure 8.12 Quantified block diagram of the interferometer	149
Figure 8.13 Sine response	150
Figure 8.14 Step response	150
Figure 8.15 Close loop simulation	151
Figure 8.16 Reference measurement of 5.2 μm step height sample	153
Figure 8.17 Measurement of 5.2 μm sample with no stabilisation	153
Figure 8.18 Measurement of 5.2 μm step height sample with stabilisation during mechanical disturbance	154
Figure 8.19 Effect of vibration compensation on a 40Hz and 323 nm peak to peak sinusoidal disturbance	155
Figure 8.20 Vibration attenuation for a range of disturbance frequencies	156
Figure 8.21 The WSI repeatability when the stabilisation is switched on	158
Figure 8.22 The interference pattern of single pixel before stabilisation of background laboratory noise	158
Figure 8.23 The interference pattern of single pixel after stabilisation of background laboratory noise	159
Figure 9.1 Sequence of operation	161

List of Tables

Table 3.1 Relation between the coherence length and spectral linewidth	38
Table 4.1 Performance targets	55
Table 5.1 Lateral ranges for different objective lenses	77
Table 5.2 The lateral resolution based on Rayleigh criterion	78
Table 6.1 Measured results for $h_r=4.707 \mu\text{m}$ step height standard using four algorithms	98
Table 6.2 Measured results for $h_r=100 \text{ nm}$ step height standard using four algorithms	99
Table 6.3 Measurement repeatability for the WSI using 4.707 μm step height sample	111
Table 7.1 Format of the created SDF file	131
Table 7.2 Parallel programming performance versus sequential programming	132
Table 7.3 Accuracy of CUDA program with respect to Matlab	133
Table 8.1 Glossary of symbols	139

Table 8.2 Comparison of repeatability when the stabilisation ON and OFF	157
Table 9.1 Specification of WSI.....	163

List of Abbreviations

AFM	Atomic force microscope
AOM	Acousto-optic modulators
AOTF	Acousto-optic tunable filter
API	Application programming interface
BS	Beam splitter
CCD	Charge-coupled device
CCI	Coherence correlation interferometry
CL	Collimating lens
CMM	Coordinate measuring machine
CO ₂	Carbon dioxide
COL	Coupling lens
CPU	Central processing unit
CSI	Coherence scanning interferometry
CUDA	Compute unified device architecture
E-system	Envelope-filter
FFT	Fast Fourier transform
FG	Frame grabber
FLOP	Floating point operation
F-operator	Form operator
GPU	Graphic processing unit
ISO	International organisation for standardisation
L-filter	Large scale filter
LVDT	Linear variable differential transformer
MEMS	Micro-electro-mechanical system
M-system	Mean-line filter
NA	Numerical aperture
NEMS	Nano-electro-mechanical system
NPL	National physics laboratory
OF	Optical fibre
OL	Objective lens
OPD	Optical path difference
PBC	Pulse to binary converter
PC	Personal computers
PD	Photo-detector
PH	Pinhole
PI	Proportional-integral controller
PSI	Phase shifting interferometry
PZT	Piezoelectric translator
R _a	Arithmetic mean of the magnitude of the deviation of the profile from the mean
R _q	Root mean square value corresponding to R _a
RC	Resistor–capacitor
RF	Reference mirror
RS _m	Mean spacing between profile peaks at the mean line

S-filter	Small scale filter
SLED	Super-luminescent light emitting diode
SPM	Scanning probe microscope
STM	Scanning tunnelling microscopes
SWLI	Scanning white light interferometry
VSWLI	Vertical scanning white light interferometry
WSI	Wavelength scanning interferometry

List of Nomenclature

E_o	amplitude of electric field.
Ω	is the angular frequency
φ	phase of the light
f	frequency
c	speed of propagation of the wave,
λ	wavelength of the light,
n	refractive index of the medium
z	distance between the origin and the wave position
I	light intensity
v	fringe visibility
I_{max}	maximum contrast of the fringe
I_{min}	minimum contrast of the fringe.
$ \gamma $	coherence function
$\Delta\varphi$	phase difference of the fringe
r	lateral resolution
NA	numerical aperture
α	phase shift
f_m	modulating frequency
OPD	optical path difference
λ_a	wavelength of light source a
λ_b	wavelength of light source b
λ_s	synthetic wavelength
a	DC bias of the fringe
b	amplitude of the fringe contrast
h	half of the optical path difference
Δn	birefringent crystal refractive index,
v_a	velocity of the propagated acoustic wave
f_a	frequency of the propagated acoustic wave respectively
θ_i	incident angle of the entrance light beam to the AOTF
L	interaction length between the acoustic wave and the optical radiation
λ_{FWHM}	linewidth of the filtered wavelength
k	wavenumber of the light source
x index	pixel number in the horizontal direction of the CCD
y index	pixel number in the vertical direction of the CCD
x	discrete data
P_d	pixel division
P_n	number of the CCD pixels in one direction
Lr_x	measurement lateral range in x direction

Lr_y	measurement lateral range in y direction
δh	resolution of the surface height
$\delta\varphi$	resolutions of the phase
A, B and C	coefficients of the quadratic equation
N	number of captured frames
F	number of fringe cycles within the interference pattern
ε	remaining fraction of the interference fringe pattern.
λ_n and λ_m	wavelengths of the first and last peaks respectively of the interference pattern
N_{pk}	total number of peaks.
$\Delta\lambda$	wavelength scanning range between λ_n and λ_m
$\delta\lambda$	step of the scanned wavelength
S_{5z}	maximum height of the topographic surface parameter,
θ_{\max}	maximum acceptable surface gradient,
β	half the maximum value of the angular angle of the objective lens
S_N	is the standard deviation
Lp	upper limit of the uncertainty
L_l	lower limit of the uncertainty
F_w	width of the frame in terms of pixel number
Row	row position of the specific threads in the thread hierarchy
Col	column position of the specific threads in the thread hierarchy
X	Displacement in the PZT
D_s	offset address of the data segment
K_p	proportional gain and T_i is the integrating time
E_3	Electric field in direction 3
m	mass of the actuator.
m_{eff}	effective mass
d_{33}	strain coefficient of piezoelectric
l	Length
A	cross-sectional area
s_{33}^E	elastic compliance
c	modulus of elasticity (Young's modulus)
S_3	strain of the PZT in direction 3.
T_3	stress of the PZT
Q_m	quality factor of the PZT
ζ	damping factor of the PZT
d	disturbance displacement
M_p	overshoot amplitude of the PZT response

1. Introduction

1.1 Overview

Ultra-precision modern manufacture widely fabricates components having micro and nano-structures as well as freeform surfaces (Luo et al., 2005, Lyons, 2007). The growing market of such components is huge and receiving great investment worldwide (Riemer, 2011). Nowadays, the use of structured surfaces is fundamental in optics, silicon wafers, hard disk drives, micro/nano-electromechanical systems (MEMS/NEMS), micro-fluidics, and micro-moulding industries since all of these industries rely critically on ultra-precision surfaces (Thomas, 1999, Maboudian, 1997). There is however a fundamental challenge to the manufacture of such surfaces, namely how to guarantee their quality and functional performance during the manufacturing process.

It is found that the functional surface influences the quality of the manufactured workpiece. It has been shown that 90% of all engineering component failures in practice are surface initiated, through mechanisms such as adhesive wear and fatigue cracking (Blunt and Jiang, 2003). With the development of surface metrology, it became clear that the functional performance of the workpiece can be guaranteed by measuring and characterising the surface. Surface metrology can also be used to control the manufacture by measuring the surface texture and micro-geometry of the workpiece (Jiang and Whitehouse, 2012).

While surface metrology continues to stimulate the development of new instruments, the challenges faced to apply those instruments embedded (on-line and/or in-process) measurement remain considerable (Vacharanukul and Mekid, 2005). The influence of the manufacturing environment, presence of lubricants and the mechanical vibration of machining tools have made embedding measurement within the context of manufacturing shop floor systems a limited factor for manufacture. Overcoming this limitation can effectively improve manufacturing throughput and make cost reductions.

In surface metrology, instrumentation is classified as being based on contact and non-contact methods. The contact method measures the surface geometries directly as exemplified by stylus techniques. The stylus usually provides a wide measurement range with fine resolution but can be destructive for softer materials and slow for areal measurement. This makes contact methods unsuitable for embedded metrology. On the other hand, non-contact methods are non-destructive. One example is optical interferometry which measures the optical path length that contains information on surface geometries. Optical techniques usually provide high measurement resolution and are considered as being more suitable for embedded measurement than contact methods because of their fast response.

The interferometer is one of the optical techniques is widely used for surface measurement. The measurement range varies from sub-millimetre to sub-micrometre values depending on the type of the interferometry and in some cases the vertical resolution can be in the sub-nanometre range. Many interferometry techniques are suitable for fast areal measurement and so have considerable potential for use in embedded metrology.

Examples of interferometry techniques are phase shifting interferometry (PSI), scanning white light interferometry (SWLI) and wavelength scanning interferometry (WSI) (Zygo, 2012, Veeco, 2012). PSI is typically limited to measurements of nano-scale surface texture due to the well-known 2π phase ambiguity problem. This limitation can be overcome by using SWLI. The measurement speed of SWLI, however, is relatively slow comparing to other interferometers due to the mechanical scanning required. WSI can be employed to measure absolute distances without the 2π phase ambiguity limitation of PSI and in a shorter time than SWLI.

In order to consider WSI as an embedded metrology instrument, it should be capable of performing surface measurement with fine resolution (e.g. sub-micrometre vertical resolution), processing large acquired data at high speed (e.g. few seconds computing time), and robust against the effects of environmental disturbance. This thesis describes the development of a WSI instrument

using a state-of-the-art acousto-optic tunable filter (AOTF) for surface measurement, the acceleration of the data processing using parallel computation with a graphic processing unit (GPU), and the stabilisation of the interferometer using an active close-loop control system together with a piezoelectric translator (PZT).

1.2 Aim

The aim of this research is to develop a novel WSI for surface measurement and investigate its potential for embedded metrology applications. The target performance of the proposed project in a production environment shall be: a vertical measurement range of the order of 100 μm with nano-scale resolution and few seconds measurement time. The WSI should require no mechanically moving parts to perform phase shifting operations. This would represent an improvement over the existing state-of-the-art methods in terms of measurement speed and range and robustness against environmental disturbances.

1.3 Objectives

- Develop a novel WSI capable of measuring micro/nano-scale surface texture using an AOTF wavelength scanning technique and a tungsten-halogen lamp as a light source.
- Investigate a suitable algorithm to analysis the interference fringe pattern and evaluate the surface texture with high resolution.
- Accelerate the process of fringe analysis by parallelising the computing process using parallel programming CUDA structures and a commercial GPU.
- Stabilise WSI actively against environmental perturbation using a closed-loop control system together with a PZT.
- Demonstrate a WSI system that integrates the techniques of wavelength scanning, computing acceleration and interferometer stabilisation through a prototype instrument.

1.4 Contribution

The work contained in this thesis makes the following novel contribution to knowledge:

- The development and demonstration of a fast WSI, using a tungsten-halogen lamp, an acousto-optic tuneable filter (AOTF) and fast charge-coupled device (CCD), capable of retrieving surface texture information within a few seconds.
- The demonstration of AOTF sourced WSI in surface metrology to resolve step height and V-groove surface structured samples.
- The introduction of a comparison study to evaluate four algorithms to analysis the interference fringe pattern obtained from WSI.
- The development and demonstration of a parallel CUDA programming structure to accelerate the computing process of fringe analysis through generating a large number of threads to execute the program code by many cores of a GPU.
- The development and demonstration of real-time active vibration compensation to stabilise the WSI using an opto-electronic feedback control system which comprises a piezoelectric translator (PZT), analogue PI controller, photo-detector and a reference single wavelength illuminator.
- Establishment of a WSI novel instrument for embedded metrology and publication of new knowledge into the research community.

1.5 Thesis Organization

This thesis is laid out as follows:

- Chapter 2 presents a brief description of the development of surface metrology. It also gives an outline of different instrumentation techniques proposed for in-process surface

measurement. Instruments based on contact (stylus and SPM) and non-contact (optical) methods are briefly described.

- Chapter 3 introduces the basic concepts of interference between two beams of light and the nature of their temporal coherence. It also gives an overview of interferometry for surface metrology. Because WSI is of particular interest, its principle of operation and background are discussed in detail.
- Chapter 4 discusses the WSI system and presents the development methodology.
- Chapter 5 presents WSI implementation and operation. This chapter provides details of the implementation of the light source and interferometer for WSI. It also describes the communication links between the console and optical environment, see also Appendix A. This chapter presents the captured data and the interference fringes obtained from the system and the lateral range and resolution of the instrument.
- Chapter 6 introduces signal processing algorithms. Four algorithms are presented to analyse the measured data captured by WSI. A performance comparison is carried out between the algorithms in term of accuracy and resolution. This chapter also demonstrates the results of measuring standard samples with multi-step height structures and a V-groove structure.
- Chapter 7 discusses the parallel programming model used to accelerate the computing process using GPU technology. This chapter introduces the basic concept of parallel programming and presents a flowchart of the compute unified device architecture (CUDA) program. The results of acceleration and computing accuracy are also demonstrated in this chapter.
- Chapter 8 investigates an active stabilisation method to compensate for environmental vibration. This chapter presents both a mathematical model and a practical setup for

compensation. This chapter demonstrates the measurement of a step sample with an applied mechanical disturbance when the stabilisation is on and off.

- Chapter 9 introduces the overall discussion of the WSI system and provides the final specification of the instrument. This chapter also gives the conclusions and proposal for future work.

1.6 Publications and Awards

The work in this thesis has produced four peer reviewed journal papers, one international patent and five conference papers. A full publication list may be found in the ‘Publications and Awards’ section at the end of this thesis.

2. Surface Metrology

2.1 Introduction

The WSI instrument to be developed will be employed for the measurement of surfaces. This chapter presents an overview of the development of surface metrology throughout the preceding century. This chapter also introduces the modern instrumentation techniques that are currently utilised for measuring surfaces. Furthermore, their applications to in-process measurement are highlighted.

2.2 Surface Metrology

Surface metrology is the science of measuring both the deviation of a workpiece from its ideal shape and the machining marks left by a machining tool on the workpiece surface. Jiang et al. (2007a, p. 2049) defined surface metrology as “*the science of measuring small-scale geometrical features on surfaces: the topography of the surface*”. The geometrical features of a surface can be classified as (Whitehouse, 2011):

- Roughness - irregularities often caused by the manufacturing process including the impression left by grinding or polishing.
- Waviness - irregularities of a longer wavelength caused by errors in the path of the tool due to, for example, vibration between the workpiece and a grinding wheel on the tool path.
- Form - very long waves caused by error in slideways.

The roughness and waviness features are known as surface texture, and it has been found that the measurement of surface texture can be used to control the manufacture and predict the functional performance of the workpiece (Jiang and Whitehouse, 2012). For instance, the surface roughness can monitor the tool wear condition and predict the failure time of contact surfaces. It is found that

the surface roughness (e.g. R_a) is sensitive to the cutting time under different cooling/lubrication conditions due to the running-in process of cutting tools (Jiang et al., 2010) where R_a is the modulus of arithmetic mean of the magnitude of the deviation of the profile from the mean. Measuring the surface texture of thin films is also vital in roll-to-roll manufacturing processes. It can characterise the defects in the films, which are directly related to the film performance and lifespan.

Surface measurement comprises three stages (see Figure 2.1). First, acquisition of information about surface topography by instrumentation. Second, decomposition of the information into primary features (form, waviness and roughness) by filtration. Finally, characterisation of the surface by parameterisation.

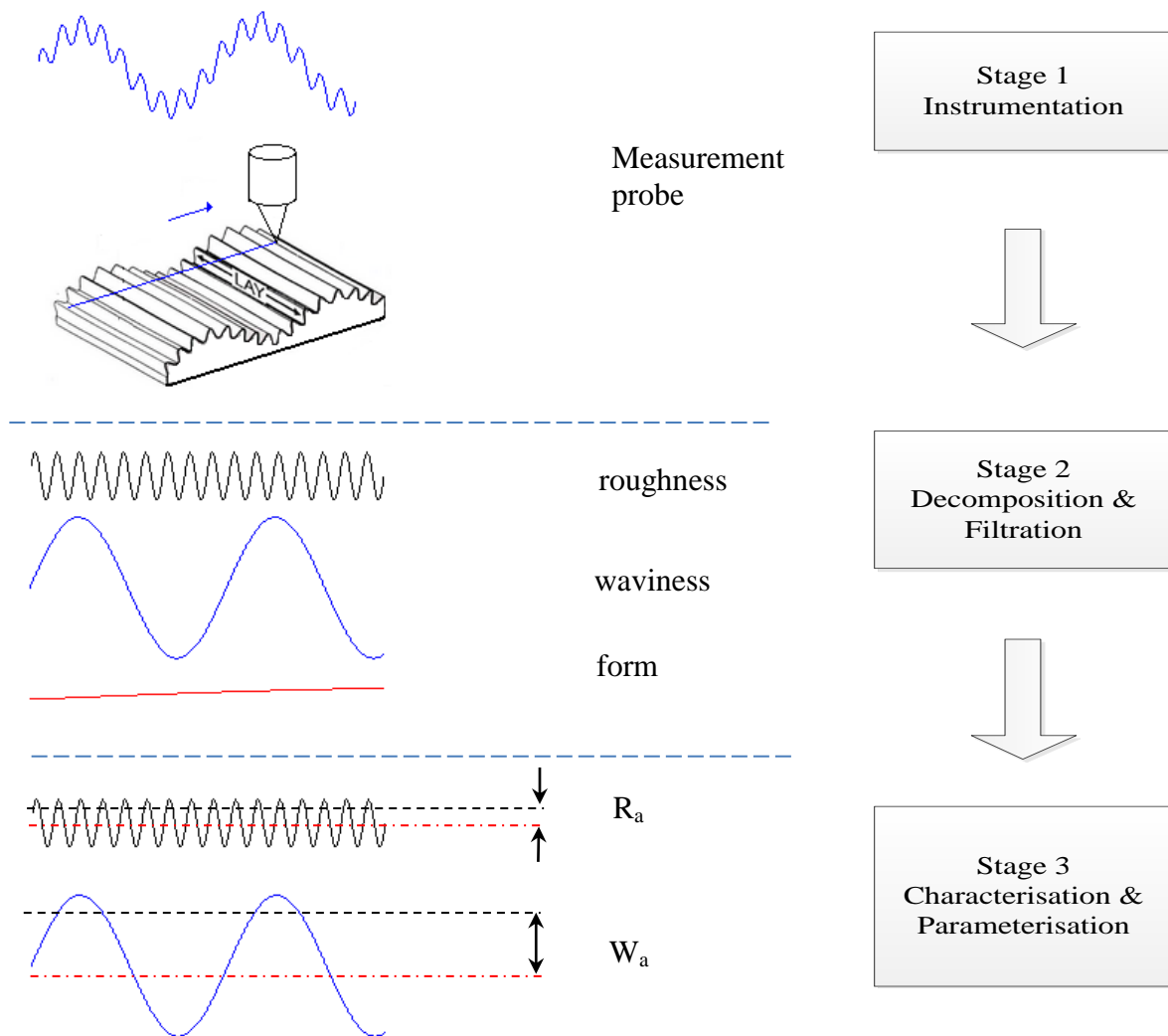


Figure 2.1 Surface measurement stages

2.3 Early Development of Instrumentation

Historically, surfaces were assessed by either a thumbnail or by eye. These subjective techniques require a person with good operating skills and experience to judge the ‘goodness’ or ‘badness’ of the surface.

Probably the first contact instrument was implemented by Tomlinson at the National Physics Laboratory (NPL) in 1919 to measure the height of the roughness (Buice, 2007). However, the first practical instruments to measure axial profile were designed in the 1930s using a stylus based method. The first non-contact optical instrument was designed by Carl Zeiss in Germany in 1934 (Whitehouse, 2011).

2.4 Filtration

To extract the geometrical features of the surface from the measured data, filters at different cutting wavelength boundaries have been developed.

The first attempts to separate the waviness from the roughness introduced two filter types; M-system (mean-line filter) and E-system (envelope filter). The M-system and E-system were an electrical RC filter and a mechanical rolling circle filter respectively.

In 1963, the phase distortion produced by the single RC filter was corrected by adding a second RC circuit. This filter was widely used in the 1960s and 70s especially after the filter was digitised at the turn of 1970. The profile produced from the 2RC filter still had some distortion but this was overcome in 1986 by introducing a Gaussian filter as a standardised phase-corrected profile filter. Further development led to ISO profile filtration toolbox which consists of five classes namely; linear filters, morphological filters, robust filters and segmentation filters (Jiang et al., 2007a).

For area filtration, the scale-limited concept is now considered. This concept is based on dividing the surface texture wavelengths into three scales namely: S-scale (contains small scale

lateral component), L-Scale (contains large scale component), and form (contains the nominal form). Each scale has its own filter: the S-filter, L-filter and F-operator respectively. The area filtration is achieved by some combination of these filters. For instance, to filter out the unwanted small-scale measurement noise and the nominal form, an SF filter can be used.

Filters such as those utilising wavelets are still under development as modern manufacturing starts turning from traditional simple geometries toward free-form geometries.

2.5 Characterisation

Surface characterisation is important for a meaningful description of the surface and an understanding of the nature of the surface. In the 1930s, an important question was whether to use a single number or a complete profile as a method of characterisation. Eventually, the development of instrumentation and filtration led to the establishment of several numerical parameters and functional curves such as the Abbott-Firestone curve (Abbott and Firestone, 1933). These parameters and curves could be successfully used in surface characterisation and help indicate the functional performance. Examples of numerical parameters used for surface profiles are amplitude parameters e.g. R_a and R_q and spatial parameters e.g. S_m (see Figure 2.2). The R_q is the root mean square value corresponding to R_a . The S_m is the mean spacing between profile peaks at the mean line (the mean of S values shown in Figure 2.2).

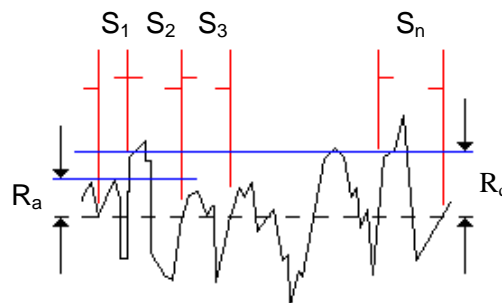


Figure 2.2 Some numerical parameters to define surface profile

The manufacturing processes developed in 1990s shifted the surface metrology from profile to areal characterisation. Areal parameters are known as field parameters and are widely used for surface characterisation. Field parameters consist of both S-parameters and V-parameters. The S-parameters describe both amplitude and spatial information. The V-parameters give volumetric information based on an areal material ratio curve (Jiang et al., 2007b).

2.6 Development of Modern Instrumentation

As stated in section 2.3 the collection of surface topography information using an instrument is carried out by either contact or non-contact methods, with each method having its own applications. This section describes the basic principle of stylus and scanning probe microscopy (SPM) contact methods as well as optical non-contact methods (interferometry, laser triangulation, light scattering and confocal microscopy).

2.6.1 Contact Method

Two types of contact methods are considered in this survey, the stylus method and scanning probe microscopy.

2.6.1.1 Stylus Based Methods

The stylus instrument is widely used for surface profile measurement. It can cover a large measurement range up to several millimetres with nanometre vertical resolution. The stylus instrument measures the geometrical features of the surface directly by a tactile method.

The stylus tip is mounted on a cantilever which is coupled to a sensor such as Linear Variable Differential Transformer (LVDT). The LVDT detects the vertical movement as the stylus traverses the sample surface at a constant speed. As such, the generated voltage represents the convolution of the stylus tip with the surface profile. The output voltage from the translator is amplified and digitized in order to be manipulated by a computer (see Figure 2.3).

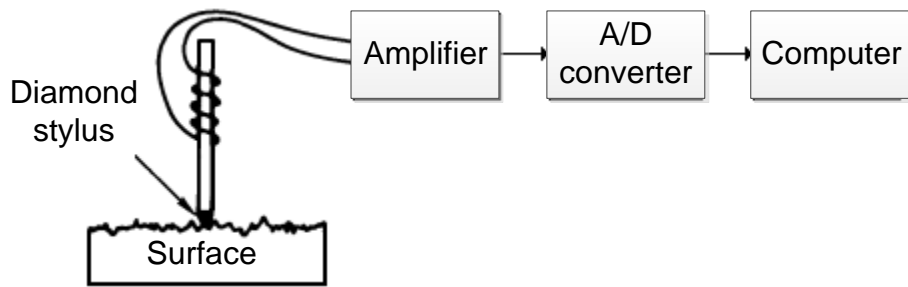


Figure 2.3 Principle of stylus instrument

The lateral resolution depends on the radius of curvature of the tip and slope of the surface irregularities (Bennett and Dancy, 1981, Elson and Bennett, 1979). For instance, Figure 2.4 shows that the sinusoidal surface sides will contact the stylus tip when either the radius of the stylus, r or the amplitude of the sample, z increases.

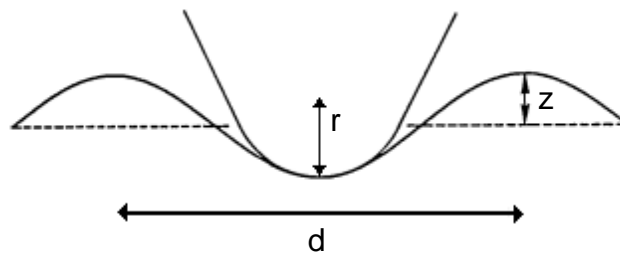


Figure 2.4 A sinusoidal surface with a stylus [adapted from Elson and Bennett (1979)]

The main disadvantage is that the stylus tip can damage the surface to be measured if the contact stress exceeds the hardness of the surface. Therefore the load of the stylus is controlled to be as low as possible such as 0.05 mg (Bhushan, 1999). Moreover, the measurement speed is relatively slow if areal surface measurement is to be undertaken.

The measurement speed and the contact approach makes stylus based methods inadequate for on-line/in-process measurement (Young et al., 1980, Bhushan, 2001).

2.6.1.2 Scanning Probe Microscopes

SPMs are commonly used for atomic scale resolution in surface measurement. The SPM has much in common with stylus instruments in the sense that a fine tip scans the sample surface to

obtain areal information. However, the main difference between the two methods is the detecting parameters (see Figure 2.5). In contrast to the stylus, where the geometry features are directly detected, the SPM detects charge density or atomic forces (Alvarez and Siqueiros, 2010). The following section describes two types of SPM probes, the scanning tunnelling microscopes (STM) and atomic force microscope (AFM).

Binnig and Rohrer (1982) published the first paper on STM. They achieved atomic-scale measurement resolution when they scanned a probe over a sample in a vacuum within a stabilised environment. The probe tip and the measured sample are brought to within angstroms of separation using a piezoelectric stage. Thus, the wavefunctions in the tip (i.e. electron clouds) overlap the wavefunctions in the sample surface. By applying a voltage between the probe tip and the sample surface, a tunnelling effect is produced (small current jumps across the tip and sample) (Binnig and Rohrer, 1983, Chen, 2008). Sensing this tunnelling current yields a very fine measurement resolution. Their work led to a Nobel Prize in 1986.

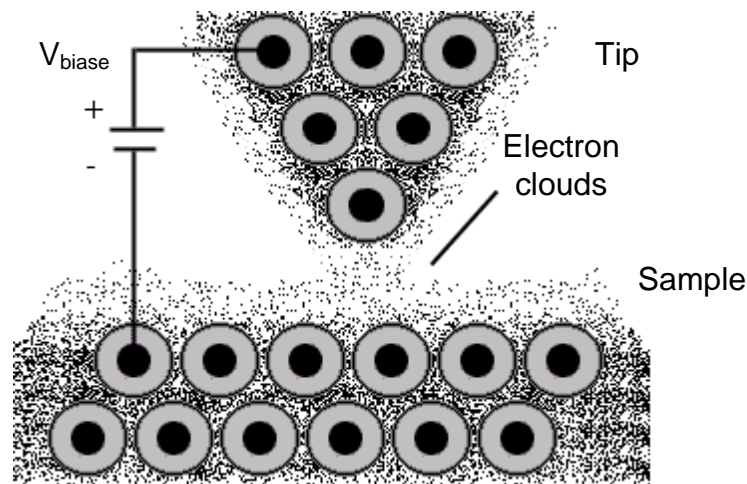


Figure 2.5 Principle of interaction between tip and sample

The STM invention extends into AFM instrumentation. The AFM basically consists of a tip of several nanometres radius of curvature attached to a micron scale cantilever. In contrast to the STM, which uses the electron clouds as a measuring parameter, the AFM uses the atomic forces that generated when the tip contacts or is in close proximity to a surface. The generated forces

results in the bending of a cantilever to which the tip is connected to. This deformation in the cantilever is usually detected by interferometry or optical deflection methods such as laser triangulation (Thirunavukkarasu, 2007).

Because of the slowness and sensitivity of the measurement procedure these types of instruments do not fit usually with the industry environment for in-process measurement applications (Bhushan, 2001).

2.6.2 Non-contact Methods

Optical methods are preferred over non-optical (e.g. capacitive methods) in surface texture measurement because of their fine vertical and lateral resolution. Also, data acquisition in optical techniques is fast which makes them suitable for on-line/in-process measurement applications. Furthermore, the non-contact concept gives them the lead for non-destructive measurement application.

The optical methods can be classified into interferometric and non-interferometric techniques. The interferometric methods are based on the detection of the interference of two beams of light. Non-interferometric are based on detecting non-interfered light reflection from a sample surface. The following sections describe some the main optical techniques.

2.6.2.1 Interferometry

Interference fringes were discovered in 1802 by Thomas Young. However, the first interferometry instrument was implemented by Albert A. Michelson in 1882. From that time up until the 1970s these instruments were used for dimensional metrology rather than surface metrology. With the development of the laser, charge-coupled device (CCD) and personal computers (PC) interferometry found a place in surface metrology. Many algorithms have been developed to analyse interference fringes and hence reconstruct surface texture.

The principle of interferometry is described in Chapter 3. Nevertheless, it is important to mention here that optical interferometry techniques are extremely sensitive to environmental noise such as mechanical vibration, air turbulence and temperature drift (Adhikari, 2004). These noise sources cause errors in surface measurement and can produce invalid results.

One of the main methods to minimize environmental disturbances in interferometers is to acquire the data as fast as possible by employing a high speed camera and fast phase shifting method. It is important to note however, that there a trade-off exists between the speed of data acquisition data and the fringe visibility which directly affects the measurement accuracy and uncertainty. Reducing the exposure time of a CCD reduces the light intensity throughput which might be critical in applications such as measuring transparent surfaces. Alternative methods suggest acquiring the required data simultaneously without reducing the exposure time (Millerd et al., 2004, Ngoi et al., 2001, Smythe and Moore, 1984).

Ngoi et al. (2001) introduced a non-mechanical phase shifting interferometer (PSI). In this method four phase shifts are achieved simultaneously by the use of quarter-wave plates and polarizers. The phase shifts are also detected simultaneously by four CCD cameras. It was found that the environmental vibration has no effect on the arithmetically averaged roughness parameter, $R_a \approx 2 \text{ nm}$, when the capturing time was 250 msec.

Non-simultaneous interferometry techniques, on the other hand, involve parts that control the impact of the environmental vibration. They work by isolating the interferometry from environmental disturbances by controlling the vibration passively by increasing the damping factor and stiffness, or by controlling the vibration using active cancellation techniques (Leach, 2000).

Controlling the environment by using vibration isolation and maintaining a fixed temperature is an effective way of reducing noise for laboratory and off-line applications (e.g. post-

process measurement), it may not be practical in the manufacturing situation, for example, when the target is too large to be mechanically isolated.

Lin et al. (2004) presented a near common-path multiplexed fibre interferometer (MFI) (see figure 2.6). This system multiplexed two Michelson fibre interferometers. One interferometer was used as a reference interferometer to compensate for environmental perturbations using a servo controller while the other interferometer was used as a measurement interferometer to measure surface profile using a dispersive technique. The reference optical path fibre circulated around a cylinder piezoelectric translator (PZT) to compensate for the unwanted vibration. Nanometre vertical resolution was obtained in a simulated manufacture environment (Dejiao et al., 2004).

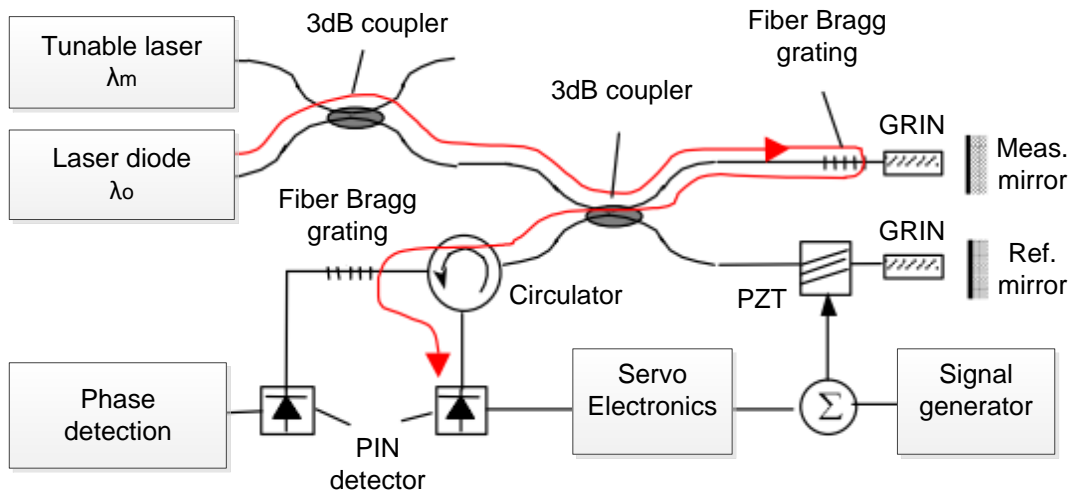


Figure 2.6 Multiplexed fibre interferometer [adapted from Dejiao, et al., 2004]

2.6.2.2 Triangulation

Laser triangulation is used in dynamic measurement to determine the distance of the surface from the photo-detector, R , then to extract the surface texture. This method is based on trigonometric relationships between incident angle of a laser beam, θ and two known distances. These are the distances of the sample surface and the worktable plane from the photo-detector, R and H respectively (Cui et al., 2010). Figure 2.7 shows the principle of operation.

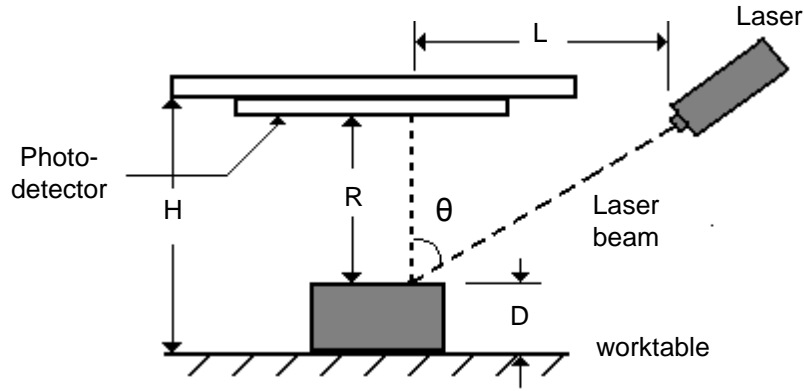


Figure 2.7 Laser triangulation method[adapted from (Groover, 2007)]

The surface height, D from the reference plane i.e. worktable, can be simply obtained by the equation,

$$D = H - L \cot \theta \quad (2.1)$$

Laser triangulation can be used in in-process measurement since the measurement response can be up to 20 kHz (MTI instruments Inc., 2012).

The resolution of this instrument is a function of the speckle contrast that appears when the surface roughness exceeds a certain level (Dorsch et al., 1994). It is found that the surface measurement resolution as high as several micrometres can be achieved for millimetre roughness scale.

2.6.2.3 Light Scattering

Light scattering techniques are widely used for applications that require rapid and non-destructive surface roughness measurement (Vorburger et al., 1993). These techniques are based on evaluating the scattering pattern obtained from a surface illuminated by a laser light (see Figure 2.8).

The scattering pattern reflected from a surface illuminated by laser light depends among other factors on the surface roughness. If the surface is perfectly smooth, then θ_i is equal to θ_r

according to the law of reflection. However, increasing the surface roughness will diffuse the light away from θ_r . By analysing the diffusion angle, the surface roughness can be obtained.

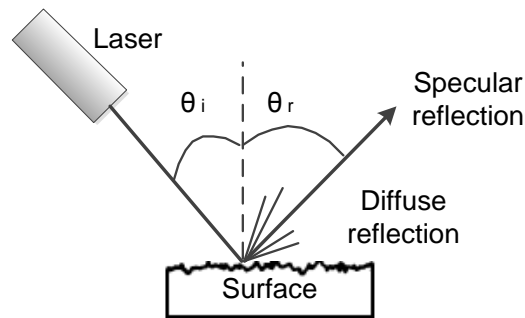


Figure 2.8 Principle of light scattering method

This method has been the subject of much research (Church and Zavada, 1975, Bennett and Mattsson, 1989). Persson (1988) presents light scattering instruments for in-process measurement of surface roughness ranges of $0.09 \leq R_a \leq 0.16 \mu\text{m}$. The instrument used to illuminate a sample while the grinding machining in operation was a He-Ne laser. The light intensity of the specular reflection was captured and analysed to determine the surface roughness. It was found that the result was of the same order as when a stylus instrument is used (Persson, 1998).

2.6.2.4 Confocal Microscopy

The enhanced image obtained using confocal microscope when compared with the conventional microscope is produced by a pinhole aperture placed in front of the detector to reject out-of-focus light (see Figure 2.9). This concept was developed by Marvin Minsky in 1955 into a working instrument (Semwogerere and Weeks, 2005).

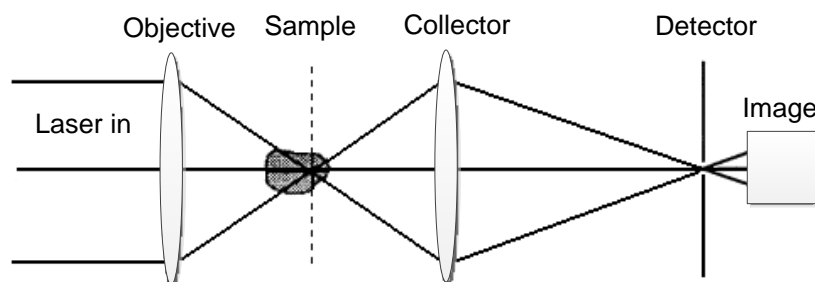


Figure 2.9 Schematic of confocal microscopy

Since only the light that focused on the sample will pass through the pinhole, slices along the axial direction need to be viewed to reconstruct areal information about the surface. This can be achieved by capturing set of images while the sample is scanned vertically (Conchello et al., 1994).

An alternative method is based on deliberately introducing axial chromatic aberration into the microscope objective to avoid the vertical scanning process. The monochromatic light source in confocal scanning microscopy is replaced by a broadband light source such as super-continuum light (Shi et al., 2004). Each wavelength in the broadband source will have its own focal plane. These focal planes all exists within the focal depth of the objective lens. This method is known as chromatic confocal microscopy.

In 1994, Tiziani and Uhde introduced chromatic confocal microscopy which has the potential to be used for three-dimensional embedded metrology measurement. The colour impression was used to evaluate the intensity distribution and to discriminate the surface height. It was found that only three images are required for areal topography which significantly improved the measurement throughput. This advantage makes the instrument suitable for online measurement applications (Tiziani and Uhde, 1994).

2.7 Comparison Between Instruments

The instruments based on contact and non-contact methods work on different physical principles. The stylus measures the surface geometry, the SPM measures either charge density or atomic forces, and optical interferometric techniques generally measure the optical path length. Although all the instruments can retrieve surface topography information, the measurement principle may decide the application. For instance besides surface measurement, film thickness can be obtained using some optical instruments; the stylus can be used to measure physical properties such as the workpiece hardness in addition to surface topography (Stanford and Hagan, 2009).

In general, the measurement speed for non-contact methods is faster than for contact methods. It can be seen from the previous section that contact methods are inadequate for embedded measurement while the non-contact method techniques are potentially suitable.

The instruments have different measurement ranges and resolution capabilities (see Figure 2.10). It can be seen from this figure that the stylus has the largest vertical measurement range with a millimetre scale and fine resolution up to 10 nm. Higher vertical resolution can be achieved with optical techniques but the lateral resolution is not comparable. The highest vertical and lateral resolution can be obtained by using AFM, but it has the smallest vertical range.

Finally, in contrast to the optical methods, the stylus could damage the surface and the measurement probe needs continuous wear condition monitoring.

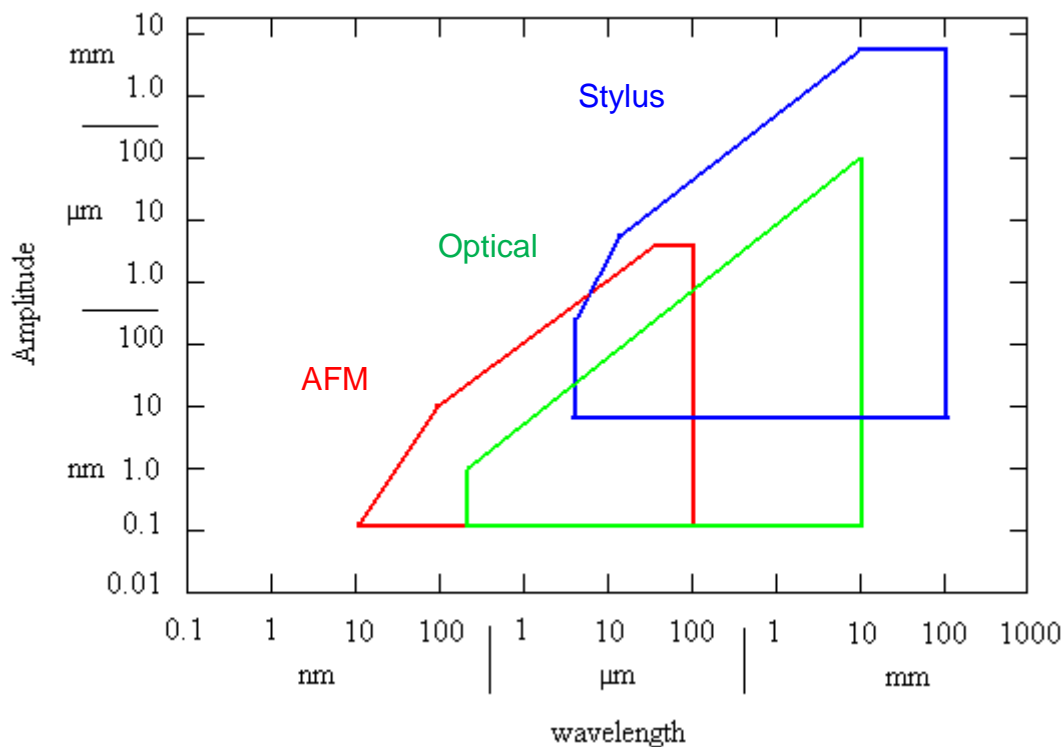


Figure 2.10 Amplitude-wavelength plot [adapted from Jiang et al., 2007b]

2.8 Summary

The geometrical features of the surface can be classified into three features: roughness, waviness and form. Measurement of these features can be used to control the manufacture process and predict the functional performance of the workpiece.

Surface metrology comprises three stages: instrumentation; filtration and decomposition; characterisation and parameterisation. The instrumentation acquires the surface topography information. Filtration decomposes the surface texture into its primary features. Finally the surface is characterised by numerical parameters and functional curves.

The instrumentation can be divided into those using contact and non-contact methods. The contact method measures the geometric features of the surface physically. It can be destructive and is considered inadequate for embedded measurement. The measurement speed is relatively slow for areal surface measurement. The non-contact method, on the other hand, measures the optical path length. It is a non-destructive method, has relatively large measurement range up to 1 millimetre and sub-nanometre vertical resolution and has much potential for embedded measurement. This method can usually acquire the measurement data at high speed.

In conclusion, optical methods, especially interferometers, are the techniques that with most having potential to be used for micro/nano-scale embedded surface metrology.

3. Interferometry

3.1 Introduction

As discussed in Chapter 2, interferometry has enormous potential in surface metrology for in-process measurement. This chapter presents basic concepts of interference and gives an overview of interferometry in surface metrology. At the end of this chapter, the wavelength scanning interferometer (WSI) is discussed in more detail.

3.2 Light Interference and Coherence

Light interference can be explained by the superposition of two waves which is well documented by many textbooks (Hecht, 2002, Liu, 2005, Saleh and Teich, 2007). In this section the principles of light interference and coherence are discussed in some detail to make the study of its interaction with an instrument clear.

Although a light wave is defined as electromagnetic radiation, only the electric field component is considered when deriving the interference pattern equation. The electric field E of the light beam at a specific point can be expressed as

$$E = E_o \cos[\omega t + \phi] \quad (3.1)$$

where E_o is the amplitude, ω is the angular frequency and ϕ is the phase. The angular frequency of the light and the phase can be defined as:

$$\omega = 2\pi f = \frac{2\pi c}{\lambda} \quad (3.2)$$

and

$$\phi = \frac{2\pi}{\lambda} nz \quad (3.3)$$

where f is the frequency, c is speed of propagation of the wave, λ is the wavelength of the light, n is the refractive index of the medium and z is the distance between the origin and the wave position. Equation 3.2 shows that light frequency is approximately 10^{15} Hz. This gives a period in femtoseconds which is much shorter than the time response of any photo-detector. The time response of the photo-detector is limited to several GHz. This yields a time averaged DC light intensity component as given in Equation 3.4.

$$I = E_o^2 \quad (3.4)$$

The interference pattern of two beams each having a same single frequency (i.e. monochromatic) can be derived using the complex exponential form method which is based on representing the cosine electric field term by the real part of the complex exponential form as given below:

$$E_1 = \text{Re}\{E_{o1}e^{i\phi_1}e^{i\omega t}\} \quad (3.5)$$

$$E_2 = \text{Re}\{E_{o2}e^{i\phi_2}e^{i\omega t}\} \quad (3.6)$$

The total electric field can be obtained by superposing the two waves as:

$$E = E_1 + E_2 \quad (3.7)$$

It then follows from Equation 3.4 that the resulting intensity is:

$$E^2 = E_{o1}^2 + E_{o2}^2 + E_{o1}E_{o2}[e^{i(\phi_1-\phi_2)} + e^{-i(\phi_1-\phi_2)}] \quad (3.8)$$

Writing the real part of the exponential form in its cosine form, the interference pattern of two monochromatic waves can be written in terms of their intensities as:

$$I = I_1 + I_2 + 2\sqrt{I_1 I_2} \cos(\Delta\phi) \quad (3.9)$$

where I is the intensity of the interference pattern, I_1 and I_2 are the intensities of the first and second beams respectively and $\Delta\phi$ is the phase difference between them (i.e. $\phi_1 - \phi_2$). This equation illustrates that the interference pattern has a sinusoidal pattern that represents the constructive and destructive interference fringes. The fringe visibility is defined as:

$$v = \frac{(I_{\max} - I_{\min})}{(I_{\max} + I_{\min})} \quad (3.10)$$

where v is the fringe visibility, I_{\max} is the maximum fringe contrast and I_{\min} is the minimum fringe contrast.

The fringe visibility is constant at any point along the optical path when the light beams are perfectly coherent (i.e. the emitted waves having the same frequency and phase such as the waves emitted from lasers). However for quasi-monochromatic light sources, the fringe visibility depends on the coherence properties of the light source. The coherence has two types, spatial and temporal.

The spatial coherence is the phase correlation between two points that lie within some finite area on a wave front. If there is no phase difference at any time during the wave front travel, then the waves have perfect spatial coherence. However, if the phase correlation has specific limits, then the wave is described as partially spatially coherent.

Temporal coherence is the time delay, $\Delta\tau$ and phase difference at a specific point between two electric fields remains constant. However, if the phase difference remains constant only at specific time delay then the wave is described as partially temporally coherent. This type of coherence directly influences the fringe visibility of interfering waves sharing the same optical axis such as those in Michelson and Linnik interferometers. This phenomenon is studied in more detail in this thesis since the Linnik interferometer will be used in the project.

The temporal coherence can also be given as a coherence length. The coherence length is the optical path difference (OPD) for which the fringe visibility remains within a specific prescribed

number. In general this represents a full-width half-maximum (FWHM) or $1/e^2$ of the coherence function. The coherence function can be obtained practically by altering the OPD and registering the corresponding drop in intensity. The envelope function represents the coherence function $|\gamma|$. As such equation 3.9 can be modified to the following form for partially coherent interference.

$$I = I_1 + I_2 + 2\sqrt{I_1 I_2} |\gamma| \cos(\Delta\varphi) \quad (3.11)$$

The coherence length is related inversely to the bandwidth of the light. For narrow spectral linewidth (λ_{FWHM}) a longer coherence length is obtained. The coherence length also depends on the spectral profile, e.g. whether it is Gaussian or Lorentzian. Table 3.1 illustrates the relation between the coherence length and spectral linewidth.

Table 3.1 Relation between the coherence length and spectral linewidth

Spectral profile	Coherence length(L_c)	Example: if $\lambda=683.42$ nm and $\lambda_{FWHM}=2.2$ nm
Gaussian	$\approx 0.42 (\lambda^2/\lambda_{FWHM})$	$L_c \approx 89 \mu\text{m}$
Lorentzian	$\approx 0.66 (\lambda^2/\lambda_{FWHM})$	$L_c \approx 140 \mu\text{m}$

3.3 Interferometry for Surface Metrology

Surface topography assessment at the nano and micron scales needs measurement instruments that can provide high vertical and lateral resolution. This is possible if microscope objective lenses are combined with interferometers so that the lateral resolution depends on the objective lens while the vertical resolution depends on the interferometer. When choosing an objective lens, trade-offs must be made between: the required field of view, the working distance and depth of focus of the objective lens, the amount of reflected light back through the objective lens, the surface gradient of the measured sample, and the required lateral resolution.

The lateral resolution depends, among other factors, on the magnification factor of the lens. High magnification lenses can provide micron and even sub-micron resolutions. For example a 10X

type objective lens gives approximately 1.4 μm of lateral resolution for an illumination source having wavelength 683 nm and an objective lens with numerical aperture 0.28. This lateral resolution is based on the Rayleigh criterion which identifies the lateral resolution as:

$$r = 0.61 \frac{\lambda}{NA} \quad (3.12)$$

where r is the lateral resolution, λ is the wavelength of the illumination source and NA is the numerical aperture.

Vertical resolution depends mainly on the interferometer, generally ranging from sub-nanometre to a few microns based on the measurement technique that is used in the interferometer e.g. phase shifting, heterodyne.

3.3.1 Interferometric objective configurations

Figure 3.1 shows the four common interferometric objective configurations: Fizeau, Mirau, Michelson and Linnik. All of these interferometers are based on the interfering of two beams to obtain an interference fringe pattern, as represented by Equation 3.11.

Each type of objective configuration has its own applications. For example the Fizeau is well known for monochromatic and dispersive interferometry because of the unbalanced optical paths configuration. This configuration requires a longer coherence length which means it is not suitable for white light interferometry. The Michelson, Mirau and Linnik interferometers are well suited for interferometry that requires best fringe contrast because they have balanced optical paths between the reference and measured surfaces. The best fringe contrast is obtained in these three configuration when the OPD=0.

The Michelson, Mirau and Fizeau interferometers place the beam splitter or the glass plate between the measured surface and the objective lens. Therefore, these interferometers are not suitable for measurements that require a long working distance. However, the Linnik interferometer

suffers no such limitation because it places the beam splitter before the objective lenses. This gives the Linnik the ability to use high NA objective lenses and to be used for measurement applications that need long working distances.

Due to the position of the beam splitter, the Linnik interferometer has a separated reference path and an extra objective lens. The separated reference path makes the interferometer more susceptible to environmental disturbances; in addition, two matched high quality objective lenses are necessary to balance the optical paths and minimise the wave front aberration.

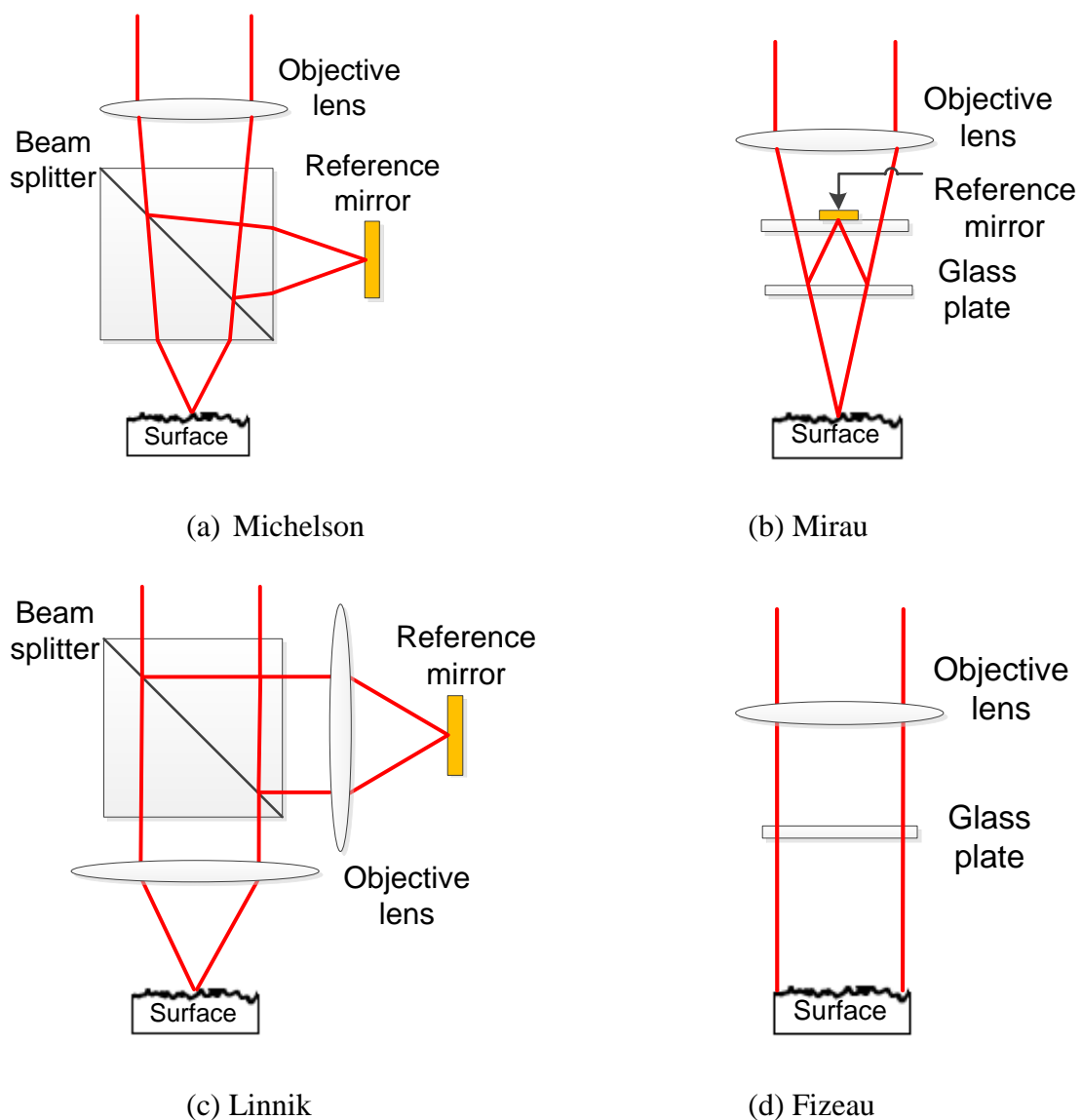


Figure 3.1 Interferometric objective configurations

3.4 Phase Shifting Interferometry

Phase Shifting Interferometry (PSI) is one of the most common and widely used techniques to measure super polished and smooth surfaces. The measurement resolution can be up to 1/1000 of a fringe under optimum conditions (Kafri, 1989).

The fundamental principle of this interferometry is to modulate (i.e. shift) the phase by physically moving one of the interferometer arms. This is often done by pushing the reference mirror using a PZT. After each phase shift the light intensity is registered. These registered intensities are then processed by a PSI algorithm in order to determine the original phase.

It is necessary for the phase to be shifted equally n times that is the mirror must be moved equal distances. This is because the interference pattern contains three unknown variables: the DC intensity bias, I_o ; the visibility, v ; and the original phase, ϕ . As such, the minimum requirement is three sets of intensity readings (i.e. $n=3$). The following mathematical description is used for the case of n phase shifts (Schreiber and Bruning, 2007).

$$\begin{bmatrix} I_1 \\ I_2 \\ \cdot \\ \cdot \\ I_n \end{bmatrix} = \begin{bmatrix} I_o[1 + v \cos(\phi + \alpha_1)] \\ I_o[1 + v \cos(\phi + \alpha_2)] \\ \cdot \\ \cdot \\ I_o[1 + v \cos(\phi + \alpha_n)] \end{bmatrix} \quad (3.13)$$

where α is the induced phase shift. The most common algorithms to retrieve the original phase value are the Carré algorithm, three shift algorithm, four shift algorithm and five shift algorithm (Wyant et al., 1984, Carré, 1966, Schwider et al., 1983, Hariharan et al., 1987). Except for the Carré algorithm, they all require a known amount of phase shift to be applied. For instance, the three shift algorithm requires a shift of 90° . In 1966, Carré introduced a technique for PSI that is independent of the magnitude of the phase shift, thus it may be an arbitrary amount. However, it has been

suggested that phase shift should ideally be 110° to reduce the influence of the random intensity noise (Kemao et al., 2000).

Various phase unwrapping algorithms are designed to correct the 2π phase ambiguity due to noise or 2π wrapping limitation of trigonometric functions (Huntley, 1989, Itoh, 1982, Charette and Hunter, 1996). Regardless of the phase unwrapping algorithm utilised, the main drawback of PSI is the 2π phase ambiguity when there is a physical OPD jump greater than $\lambda/2$ (Schreiber and Bruning, 2007). This issue limits the applications of PSI.

3.5 Heterodyne Interferometry

Heterodyne interferometry is a high precision technique which may be used to measure the optical path difference to within nanometre accuracy. The concept of this interferometric technique is to introduce a frequency shift between two interferometer beams. The two beams are mixed at a photo-detector to obtain a modulated electric signal. Since the photo-detector has a nanosecond response time which is much longer than the period of the laser waves, the obtained electric signal contains just the shifted frequency (i.e. modulated frequency) the phase of which corresponds to the OPD as expressed in the following equation

$$I(t) = I_R + I_M + 2\sqrt{I_R + I_M} \cos(2\pi f_m t + \Delta\phi) \quad (3.14)$$

where f_m is the modulating frequency and $\Delta\phi$ is the phase shift that contains the OPD information. The modulating frequency is usually induced by two acousto-optic modulators (AOM). The frequency difference between the two AOM frequencies is the modulating frequency (Hariharan, 1991).

The modulating frequency is usually set from tens of kHz to a few MHz, and the phase is then be tracked electronically. Xiaoli and Katuo (1998) proposed an interferometric method which contained the heterodyne technique to measure the excess phase fraction. Two AOMs were

included in the setup with driving frequencies 80 MHz and 80.1 MHz. The detected modulating frequency was 0.1 MHz and the phase then tracked by a lock-in amplifier. The obtained result has nanometre precision.

The main drawback of this type of interferometry is the 2π phase ambiguity which limits the measurement range of the OPD to a half wavelength of the light source.

3.6 White Light Interferometry

Broad band light sources are widely used with optical interferometers. In contrast to monochromatic interferometry, the visibility of the fringe in white light interferometry is generally limited to a few microns. This is because the spacing of the interference fringes depends on the wavelength, except when the OPD is zero (Wyant, 2002). Thus, the maximum visibility occurs when there is no OPD between the measured and reference paths, whereas the visibility is reduced when the OPD increases as shown in Figure 3.2 (Balsubramanian, 1980). This phenomenon is combined with vertical scanning techniques to produce the Coherence Scanning Interferometer (CSI) (Badami et al., 2012).

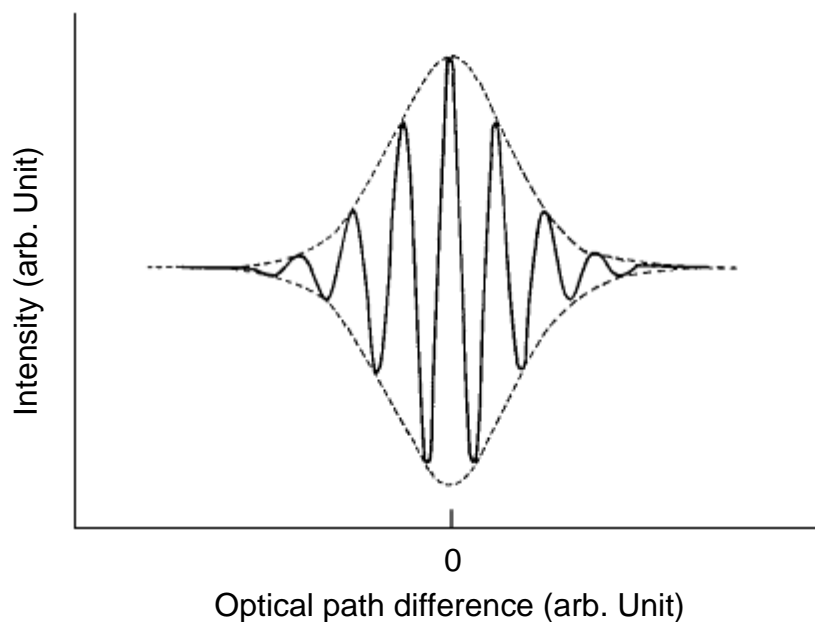


Figure 3.2 Interferogram for scanning white light interferometry

The CSI is widely used to assess surface topography by analysing the coherence function obtained from the interfering light beams (Petzing et al., 2010). Since the white light has a short coherence length, the maximum variation in the fringe visibility can be easily monitored by capturing a sequence of surface images when the reference mirror is scanned mechanically. This interferometer approach is also known by several other names such as vertical scanning white light interferometry (VSWLI) and coherence correlation interferometry (CCI). Figure 3.3 shows the Talysurf CCI 6000 produced by Taylor-Hobson Ltd. The vertical resolution of this instrument is 0.01 nm and the standard measurement range is 100 μ m (Taylor-Hobson, 2005).

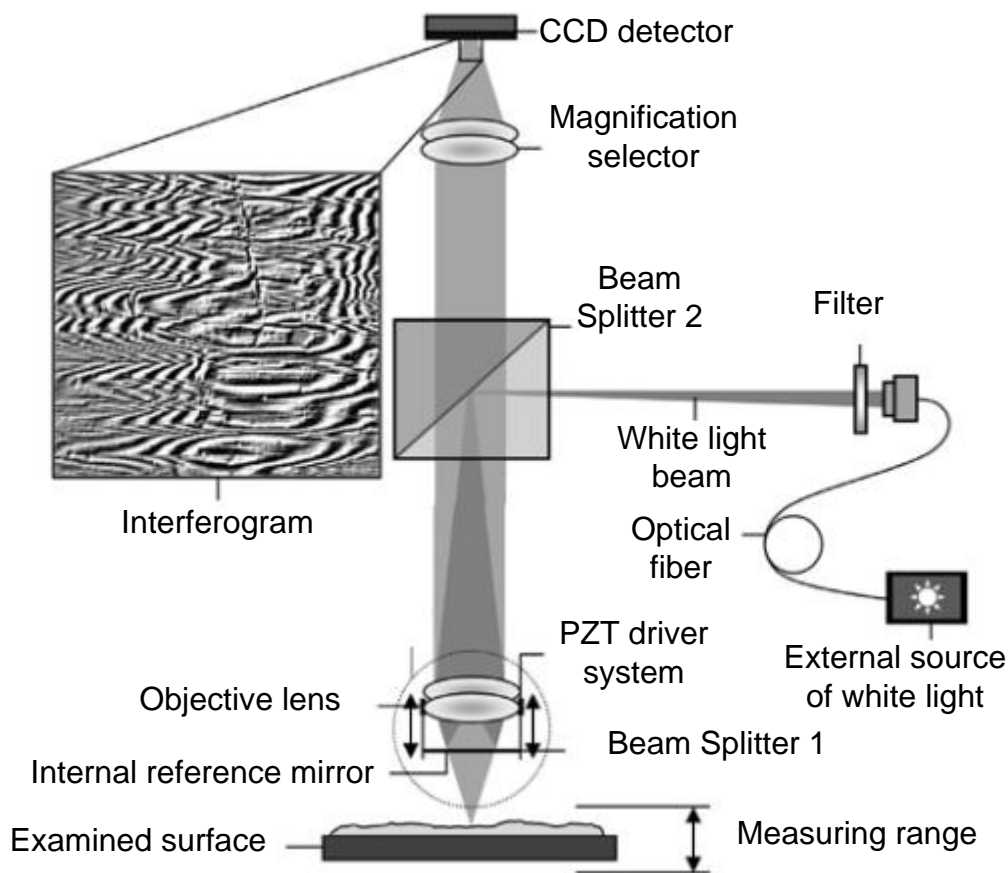


Figure 3.3 Schematic of CCI 6000 [adapted from (Kaplonek and Lukianowicz, 2012)]

The measurement range is limited only by the scanning transition stage which could be varied from hundreds of micrometres for PZTs to tens of millimetres for stepper motors.

Although highly accurate measurements can be obtained using scanning white light interferometry, measurement errors such as “bat wings” may occur when the OPD is less than the coherence length of the broad band light (Gao et al., 2008, Wyant, 2000). Such errors are caused by diffraction effects at the edges of step discontinuities which shift the peak position of the coherence function. However, peak shift detection and fractional phase determination from the position of the best-focus frame have been introduced by many researchers to enhance the measurements (Groot et al., 2002, Harasaki et al., 2001, Larkin, 1996, Sandoz et al., 1997, Wyant, 2000). It is also found that the mechanical scanning of a heavy probe head or the specimen stage induces measurement error due to some piezo-mechanical performance interactions due to non-linearities and hysteresis errors (Kang et al., 1999). This mechanical scanning could also limit the measurement speed.

3.7 Two-Wavelength Interferometry

Conventional monochromatic interferometers suffer from a 2π phase ambiguity. Although a phase unwrapping process can be used to solve the discontinuity in the phase distribution by adding a number of 2π steps, the measurement range is still limited by the 2π phase ambiguity if the optical path difference between two adjacent pixels changes by more than half a wavelength.

To extend the measurement range, a synthetic wavelength can be constructed by using two light sources having wavelengths, λ_a and λ_b . This technique can effectively extend the measurement range.

In a two wavelength interferometer, the fringe patterns of λ_a and λ_b are added together at the image plane, this addition yields a new interference pattern similar to the beating signal. Figure 3.4 shows a simulated example where $\lambda_a = 682.42$ nm and $\lambda_b = 590.98$ nm.

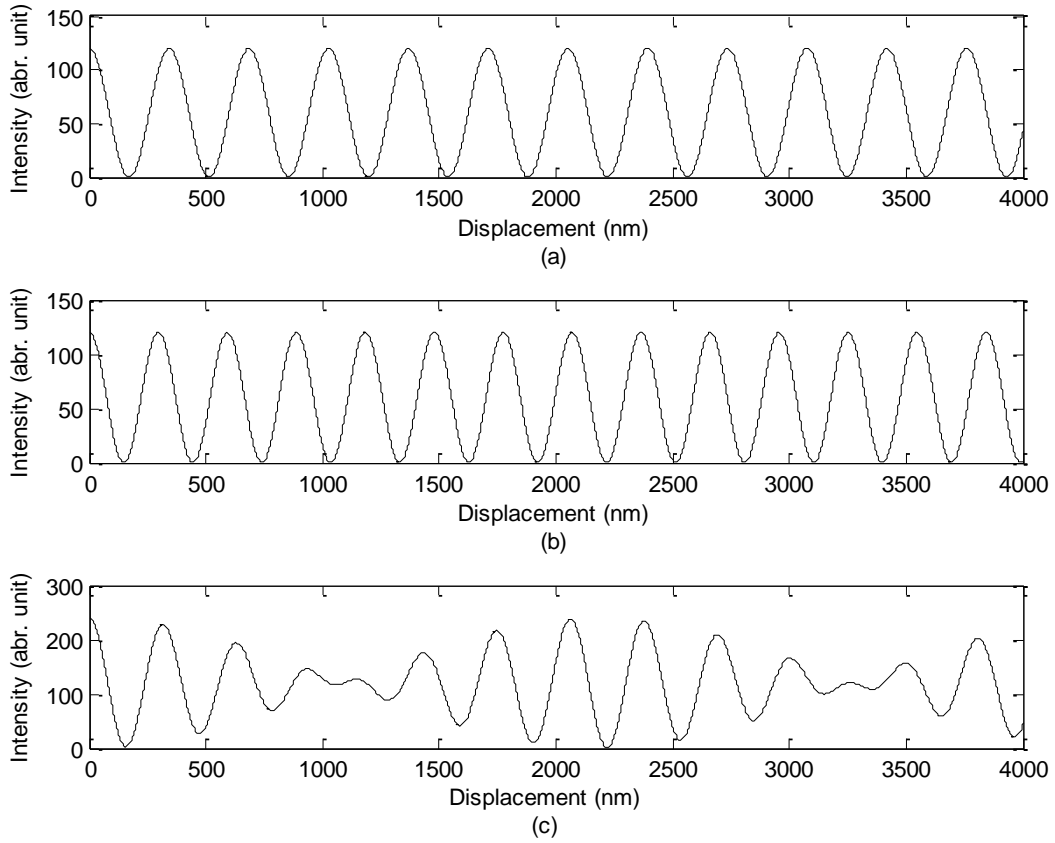


Figure 3.4 Fringes of two wavelength interferometer (a) fringes of $\lambda_a=682.42$ nm (b) fringes of $\lambda_b=590.98$ nm (c) beating signal at λ_s

The envelope of this beating signal acts as a fringe pattern for the synthetic wavelength λ_s which is:

$$\lambda_{syn} = \frac{\lambda_a \lambda_b}{|\lambda_a - \lambda_b|} \quad (3.15)$$

By analysing the envelope signal, the optical path difference can be determined. The measurement resolution of this method ranges from several to hundreds of micrometres depending on the magnitude of the synthetic wavelength.

A CO₂ laser has been used to produce the two wavelengths $\lambda_a=10.6$ μm and $\lambda_b=9.6$ μm because of its ability to oscillate on different transitions lines (Svelto, 1998). This laser is used for

two wavelength and heterodyne interferometers to measure large absolute distances such from 25 m to 100 m with an accuracy reaching to 100 μm (Matsumoto, 1986, Walsh, 1987). The two wavelength interferometer is used to extend the measurement range while the heterodyne interferometer is used to enhance the measurement resolution.

The two wavelength interferometer is also used to measure micro-structure step samples using a tuneable laser diode with a central wavelength of $\lambda=830\text{ nm}$ and double modulation technique (Chang et al., 1999). The laser intensity is modulated by altering the injection current sinusoidally and the optical path length is modulated by using a PZT. The interferometric signal is demodulated electronically using lock-in amplifier technique. A micro step structure of 100 μm was measured. The minimum detectable displacement was found to be 20 μm .

The implementation of the synthetic wavelength technique has found its way into the surface metrology for on-line surface measurement when the two wavelengths interferometer is stabilised using multiplexed fibre interferometry (Dejiao et al., 2004).

The main drawback of this method is the phase ambiguity problem that still exists when the optical path difference exceeds half of the synthetic wavelength. Furthermore, the resolution of the two wavelength interferometer is related inversely with the synthetic wavelength. Therefore, to achieve accurate measurement, another interferometry technique is needed.

3.8 Wavelength Scanning Interferometer

Wavelength Scanning Interferometry (WSI) is an interferometric method usually used to measure large absolute distances without any 2π phase ambiguity problems. The WSI measures large discontinuous step distances ranging from micrometres to meters without mechanical scanning (Kikuta et al., 1986, Thiel et al., 1995, Yang et al., 2007).

The WSI consists of tuneable wavelength light source, an interferometer in a Linnik configuration and a CCD camera (see Figure 3.5).

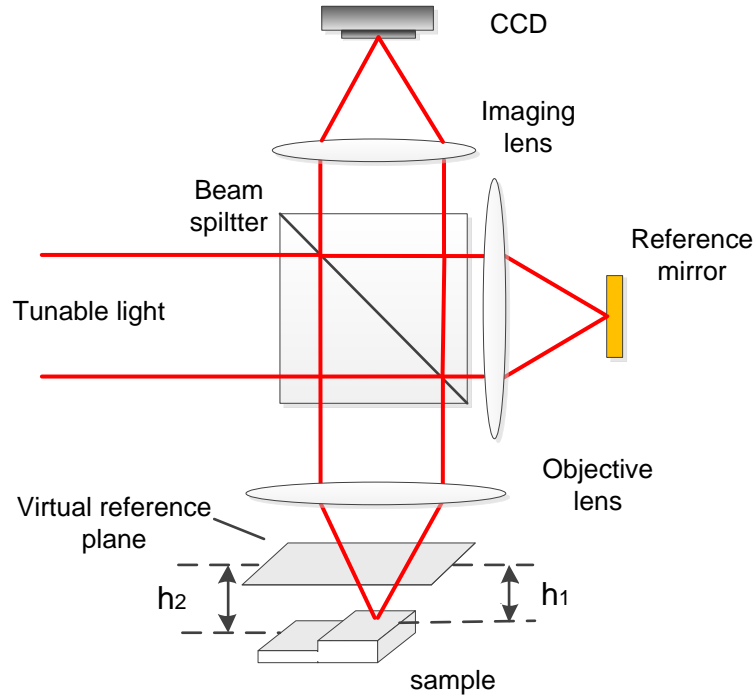


Figure 3.5 Schematic of unbalanced WSI

In contrast to two wavelength interferometry, WSI usually tunes the wavelength of a broad band light continuously and in a sequential manner. As such, no envelope interference function is obtained but instead a periodic interference pattern is produced due to the wavelength scanning process according to the intensity of the interference pattern equation:

$$I_{xy}(\lambda_i) = a_{xy}(\lambda_i) + b_{xy}(\lambda_i) \cos\left(\frac{4\pi}{\lambda_i} h_{xy}\right) \quad (3.16)$$

where λ_i is the tuned wavelength, x and y indices are the pixel number in the horizontal and vertical directions respectively of the CCD detector, a is the interference DC bias, b is the amplitude of interference contrast variation and h is the path difference between corresponding point on the reference plane and sample.

Since no mechanical scanning is performed, h is fixed at a specific point. The frequency of the interference pattern is dependent on the height of the sample with respect to the virtual reference plane. An example of the frequency change at two different heights is shown in Figure 3.6. This

figure simulates the interference pattern of two heights at a fixed wavelength scanned range (682.42 - 590.98 nm).

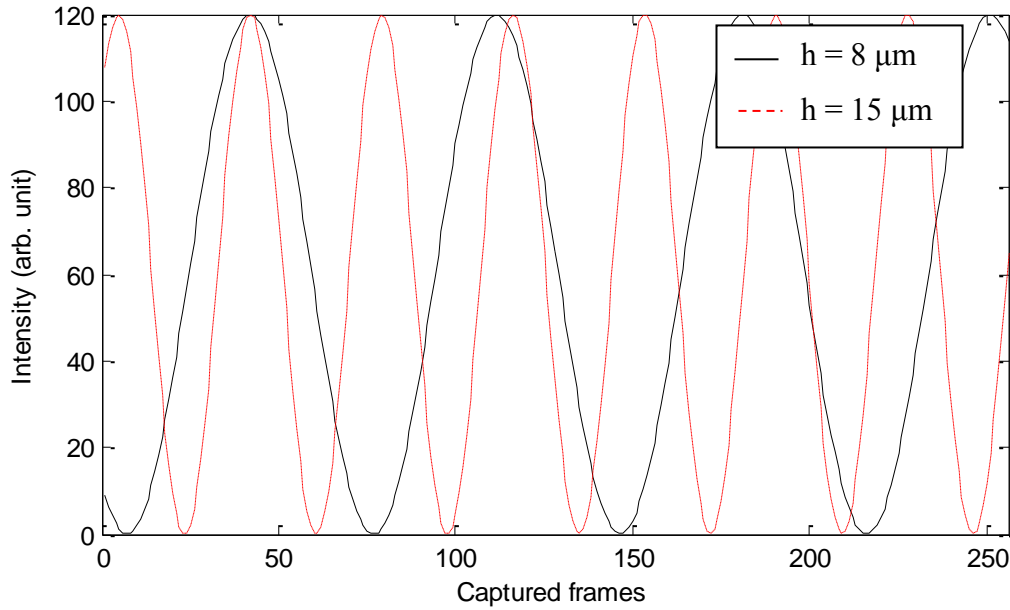


Figure 3.6 Simulated interference patterns

The evolution of the WSI is related directly to the evolution of tuneable lasers. From the 1980s to the present, the laser diode has been used as the main scanning light source for this type of interferometry. The scanning process is achieved mainly by controlling the injection current (Ishii, 1999). The tuning range of the laser diode is usually several nanometres.

The limited tuning range impacts negatively on the WSI accuracy, which induced researchers to combine WSI with another technique such as PSI to improve accuracy (Kato and Yamaguchi, 2000) or the single-wavelength heterodyne interferometer (Xiaoli and Katuo, 1998, Lu et al., 2002). Xiaoli and Katuo (1998) used WSI to identify the integer number of the fringes accurately, which is not possible in the conventional monochromatic interferometer. Wavelength scanning from 780-790 nm uses a commercial external cavity laser diode, but to identify the fraction of the fringe accurately, heterodyne technique in the same interferometer setup was used. By combining this information, a 3 mm absolute distance was measured with 3 nm accuracy.

Reports in the literature have made clear that the narrow scanning range and the mode hopping problems which exist in laser diodes limit their measurement accuracy (Thiel et al., 1995, Kikuta et al., 1986, Yamaguchi et al., 1998, Ohba et al., 1990).

In early stages of WSI development, laser mode hopping (sudden jumps in the optical frequency of a laser) was one of the major problems limiting the use of lasers as a tuneable light source. To overcome this issue, an external cavity laser diode can be used as a tuneable wavelength light source for WSI (Cabral and Rebordão, 2007). An alternative method to overcome the uncertainties of the scanned wavelength was proposed by Stone et. al (1999) based on counting the fringes produced by two separate interferometers sharing the same tuning light source. In this method, it is not necessary to identify the wavelength scanning range; hence the mode hopping disturbance that occurs in laser diodes can be overcome. This concept is explained by the following mathematical description. Equations 3.17 and 3.18 give the changes of phase during the scanning process in the measuring interferometer and reference interferometer respectively.

$$\Delta\varphi_{meas} = \frac{2\pi}{\lambda_s} OPD_{meas} \quad (3.17)$$

$$\Delta\varphi_{meas} = \frac{2\pi}{\lambda_s} OPD_{ref} \quad (3.18)$$

Since the two interferometers sharing the same synthetic wavelength λ_s and the OPD_{ref} is known a previous calibration, absolute distance in the measurement interferometer can be easily calculated using Equation 3.19.

$$\frac{OPD_{meas}}{OPD_{ref}} = \frac{\Delta\varphi_{meas}}{\Delta\varphi_{ref}} \quad (3.19)$$

A bidirectional fringe counter is used to detect the phase change in both interferometers. In addition, an AOM is placed before the two interferometers to provide a heterodyne technique for

high resolution measurement. The measurement uncertainty was found to be 0.25 μm at 5 m (Stone et al., 1999). The drawbacks of this method are the necessity for regular calibration of the reference interferometer. Furthermore, the system is susceptible to environmental disturbance because the two interferometers are separated and do not share the same optical path.

During the last decade, wide wavelength scanning ranges have been used to improve the measurement accuracy. Stabilised lasers such as dye and titanium-sapphire lasers can provide scanning ranges as high as hundreds of nanometres (Davila et al., 2012, Kuwamura and Yamaguchi, 1997, Yamamoto et al., 2001).

Kuwamura et al. (1997) introduced a WSI system for on-line measurement. This system used a dye laser as a tuneable light source for the WSI with a tuning range of 25 nm. The imaging detector was a 1D array of pixels which performed a surface profile measurement. Time domain signal processing was performed at each pixel in real time to evaluate the interference signal, which contained 256 sampling points. A zero-crossing detection technique was applied to find the 2π phase jumps during the wavelength scanning process. By identifying the wave number of the obtained zero crossing points, the OPD was determined. The system measured a 0.1 mm step height sample with 4 μm precision.

Yamamoto et al. (2001) presented a high resolution WSI using a 100 nm tuning range in the near infrared region using a titanium-sapphire laser. This research introduced a signal processing algorithm that improved the phase shift calculation. The phase shift was estimated by fitting three points around the Fourier transform peak in order to estimate the actual peak position. Areal measurement was achieved by capturing a sequence of interferograms using a 2D CCD camera. The obtained accuracy was about 0.15 μm over a 10 μm measurement range.

Takeda and Yamamoto (1994) analysed fringe patterns produced by a wavelength scanning technique using the Fourier transform. Their method was based on extracting the phase from the fringe patterns by filtering out the DC interference term and the phase conjugate using the power

spectral density which was obtained by applying the discrete Fourier transform. This method had been published twelve years before by Takeda but for a wedge type interferometer. The results of this algorithm showed that high resolution measurement better than $\lambda/30$ can be achieved (Takeda and Yamamoto, 1994, Takeda et al., 1982).

Other signal processing methods have been proposed to enhance the measurement accuracy to sub-micrometre values such as sinusoidal function fitting (Kang et al., 2001).

3.9 Summary

This chapter has introduced the basics of light interference and coherence types. Interferometry is based on analysing the fringes produced by superposition of the beams which share a source, propagation axis and plane. The interference occurs within temporal and spatial coherence ranges.

This chapter gives the basic configuration of interferometric objectives. It can be seen that Linnik interferometer can provide the largest working distance since the beam splitter is located before the objectives. This allows the use of high NA objectives with short working distances.

This chapter also introduced also several types of interferometry that are applicable to the measurement of surface topography. For super polished surfaces PSI is suitable for high precision measurement. For rough surfaces, when the optical path difference exceeds $\lambda/2$, CSI, two wavelength interferometry and WSI can be used. However, two wavelength interferometry still suffers from the phase ambiguity if the optical path difference exceeds $\lambda_s/2$.

CSI is widely used for surfaces measurement, having a large measurement range with sub-nanometre resolution. WSI is an alternative method to measure large discontinuous surfaces, but requires no mechanically moving parts to perform phase shifting operations.

The key principle of WSI is to shift the phase of the interfering beams by scanning the wavelength. This has the advantage that it can overcome limits imposed by mechanical

dynamics, such as hysteresis and non-linearity, which are inherent in any mechanical scanning method.

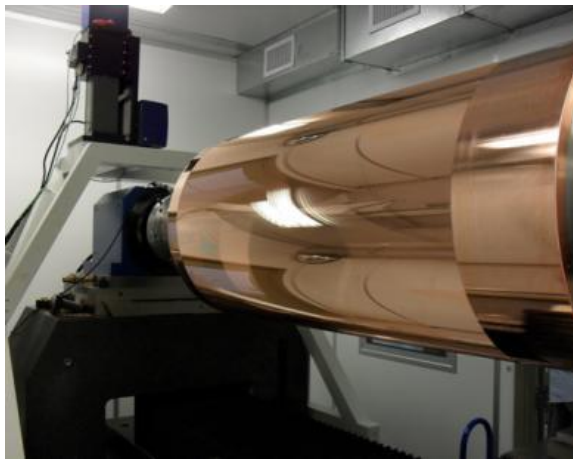
It can be seen from Section 3.8 that to implement WSI there are two main requirements: a stable wide wavelength scanning light source and a suitable phase shift determination algorithm.

4. Development of WSI instrument

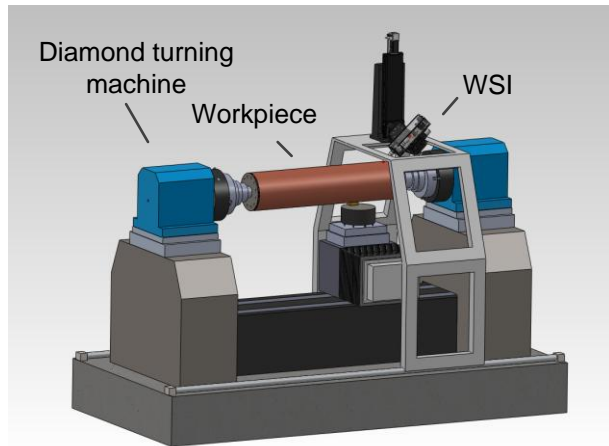
4.1 Introduction

This thesis describes the development of a fast WSI instrument for on-line/in-process surface measurements. The overview of surface metrology in Chapter 2 and the background to interferometry in Chapter 3 made clear that interferometry can measure to micro/nano-scale structures on surfaces. It is also true however, that the interferometer needs to be stabilised for embedded measurement.

One of the main applications for this instrument is to measure the surface texture of a master drum that used for roll-to-roll film manufacturing (see Figure 4.1). The master drum is usually 1-2 meters in length, and manufacturing costs make the off-process inspection impractical. For in-process measurement, WSI can be considered for use with these drums if the system is stabilised and the analysis of the measured data is accelerated.



(a)



(b)

Figure 4.1 Example of measurement application (a) Master drum [adapted from (UPS2, 2011)] (b) WSI on the machine

4.2 System Scheme

The current industry requirement for embedded surface metrology is an interferometer with a large measurement range, fine vertical and lateral resolution, robustness against environmental disturbances and fast computational time. Thus, the performance targets for the proposed interferometer are stated in Table 4.1.

Table 4.1 Performance targets

Performance target		Value
Measurement range		> focal depth of the objective lens (e.g. > 92 μm for 2X objective)
Vertical resolution		< 30 nm
Lateral resolution		Diffraction limited
Measurement time	Capturing time	< 3 sec
	Analysis time	< 2 sec
Stabilisation bandwidth		> 200 Hz

In order to develop an interferometer with the specified performance targets, the design shown schematically in Figure 4.2 was developed. This layout is based on using a WSI type interferometer as a measurement probe. The console is used to analyse the interference fringes and control hardware elements such as driving the AOTF. The stabiliser uses active servo control system to compensate for alteration in optical path difference caused by environmental disturbances.

The light source of the system combines two illumination sources working at two different wavelength regions (visible and infrared). The visible illumination source consists of a tungsten-halogen lamp and an acousto-optic tuneable filter (AOTF). The infrared source is a super-

luminescent light emitting diode (SLED). The AOTF is used to scan a filtered narrow light band for measurement purposes while the SLED is used for stabilisation purposes. The interferometer is illuminated simultaneously by the two illumination sources. With such a setup the interferometer acts as two interferometers sharing the same optical path. One is used as a measurement probe for the WSI and the other as a feedback sensor to stabilise the interferometry.

The measurement data produced by the WSI probe is captured by high speed CCD. The data is transferred to the console in order to evaluate the sample surface. The signal processing is performed by a graphic processing unit (GPU) to achieve high computing speed. In contrast to conventional single thread operations in a central processing unit (CPU), the GPU can process massive numbers of threads using many processing cores. This computing acceleration process can achieve high overall measurement throughput and make the instrument suitable for embedded measurement. The console is also used to control the system in terms of the AOTF scanning process, capturing sequence of the CCD and switching the stabilisation action.

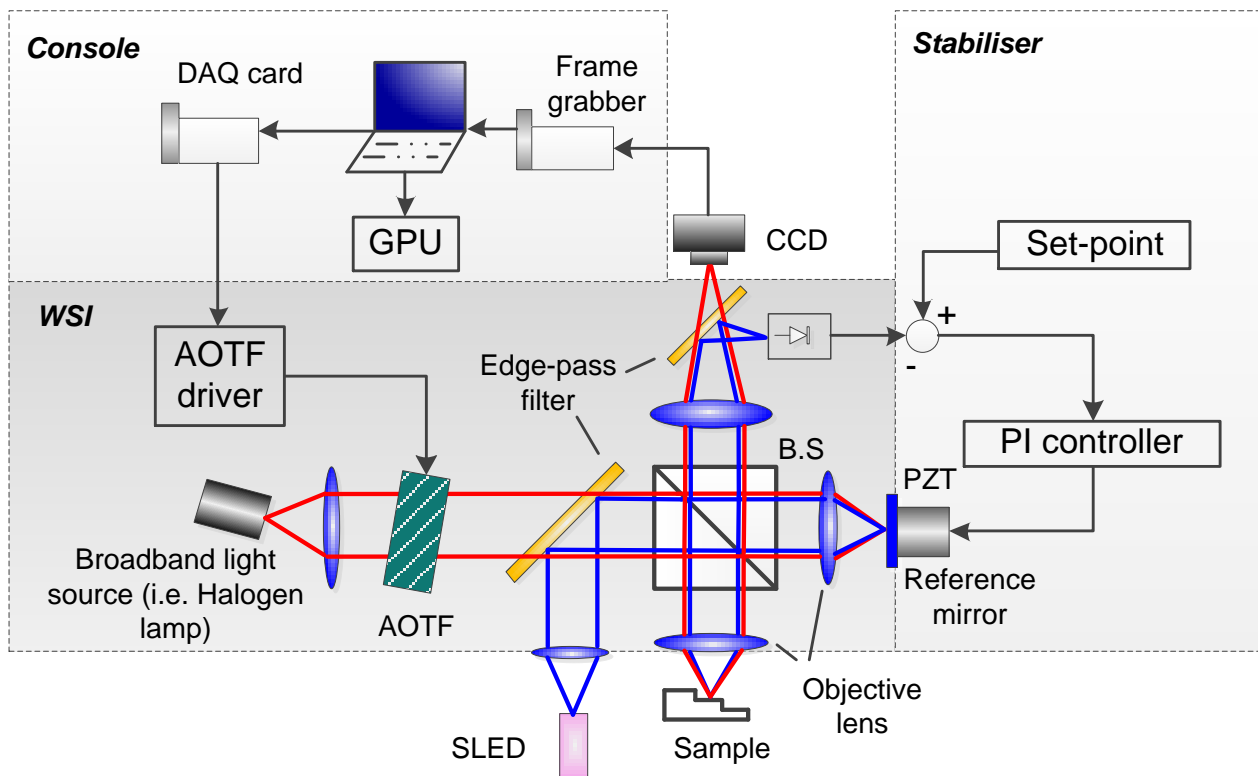


Figure 4.2 Schematic of the WSI interferometry system

4.3 Development Methodology

According to the scheme described above, the required measurement range and resolution can be achieved using a WSI. The computational time can be accelerated by using the GPU. The necessary robustness can be obtained by actively stabilising the interferometry. Therefore, the development of the instrument is divided into three phases as shown in the Figure 4.3.

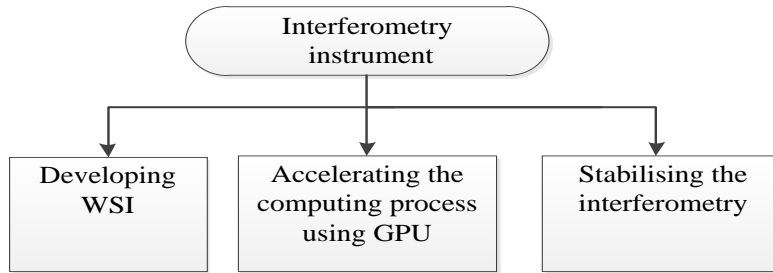


Figure 4.3 Development strategy

4.3.1 Development of WSI

The development of the WSI consists of acquiring a suitable light source, constructing the interferometer and analysing the measured data. These procedures are introduced in Chapters 5 and 6. The light source must have a wide scanning range, linear scanning behaviour and coherence length exceeding the focal depth of the objectives lenses to be used. These requirements are achieved by using an AOTF with tungsten halogen lamp. The light source performance is investigated later in Chapter 5.

The interferometer demands a long working distance and high lateral resolution. The Linnik interferometric configuration was chosen to satisfy these requirements. The lateral range and resolution mainly depend on the numerical aperture of the objective lens and the light wavelength. The implementation steps, lateral range and resolution are investigated in Chapter 5.

Analysing the measured data and evaluating the surface requires the selection of a suitable algorithm to achieve nano-scale vertical resolution. Four algorithms are investigated in Chapter 6, which also includes discussion of the measurement results obtained.

4.3.2 Acceleration the computing process

Computation using traditional sequential code execution can be time-consuming which is a problem for embedded measurement applications. The computing processes can be accelerated by executing the code in a parallel manner by generating massive numbers of the threads processed by many cores of a GPU. This solution needs the selecting of a suitable parallel programming model and managing data transfer between the central processing unit (CPU) and the GPU. The system chosen uses compute unified device architecture (CUDA) as a programming model and is reported in Chapter 7.

4.3.3 Stabilisation of the interferometry

System stabilisation is vital to achieve nano-scale resolution in an industrial environment. Environmental disturbance is compensated by using a PZT combined with a PI controller. A real time feedback signal is obtained from the SLED illuminated interferometer which acts as the feedback sensor. This part of the project is developed in Chapter 8.

4.4 Summary

The scheme for the WSI instrument is outlined. The scheme consists of the WSI measurement interferometer, the stabiliser to stabilise the WSI and the console to analyse the captured data and control the operation of the system. The development of the WSI to measure surfaces to the required accuracy and speed will consist of three stages. The first stage is to implement a suitable light source, align a Linnik interferometer and analyse the measured data by a suitable algorithm. The second phase is to develop a CUDA program to accelerate the computing process using a GPU. The third phase is to develop an active method to stabilise the WSI using a PZT and PI controller.

5. Wavelength Scanning Interferometry: Implementation and Operation

5.1 Introduction

This chapter introduces the characteristics of the WSI, including its implementation and operation. This interferometer uses an AOTF to scan the wavelength across the visible region. The wavelength scanning range, spectral linewidth of the filtered wavelength, coherence length and the behaviour of the interference pattern are investigated in this chapter.

5.2 The WSI Overview

The WSI measurement probe consists of a wavelength tuneable light source, an interferometer and a computing environment (see Figure 5.1). The tuneable light source, shown in block 1, provides the wavelength scanning process needed to produce a linear phase shift. The interferometer, shown in block 2, produces interference fringes which contain the surface height information. The computing environment, shown in block 3, stores the captured interference fringes and drives the AOTF via electronic hardware.

The following sections present the WSI parts and implementation. The WSI operation and data collection are also introduced. The analysis of the captured data is introduced later in Chapter 6.

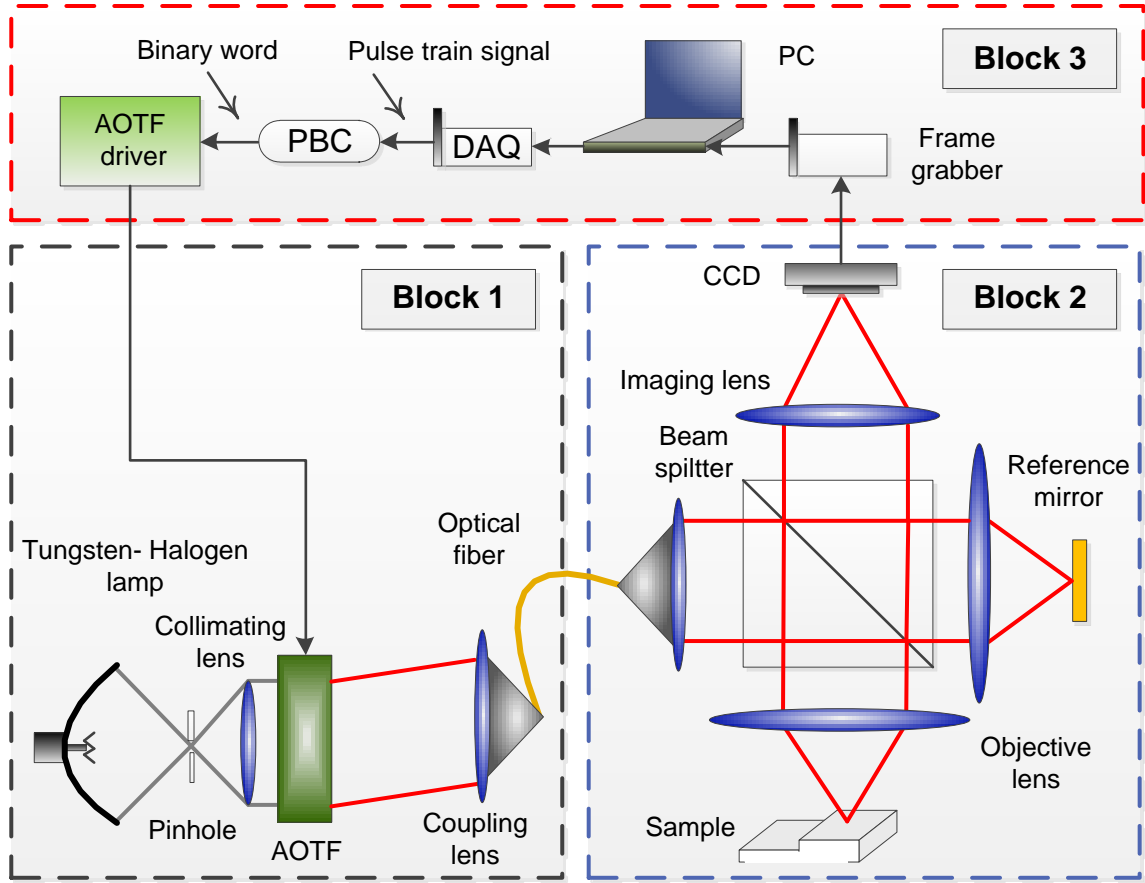


Figure 5.1 Basic configuration of the proposed WSI; where DAQ: data acquisition card and PBC: Pulse to binary converter.

5.3 Light Source Setup

It can be seen from Chapter 3 that the light source is the key feature of the WSI since the phase shift is induced by the wavelength scanning process. The main features of the light source are: (a) the scanning range, wider scanning range can enhance the measurement resolution; (b) the scanning linearity effects on the phase shift, linear scanning of the wavenumber (reciprocal of the wavelength) can produce periodic sinusoidal interference pattern; and (c) the spectral linewidth, fine spectral resolution can enhance the temporal coherence of the light source and enhance the measurement accuracy. Using a halogen-tungsten lamp combined with an AOTF as a light source instead of tuneable lasers can avoid spurious interference fringes caused by stray reflections of long coherence length laser beams.

The AOTF is used to filter a narrow spectral band from a broad band light source, providing a linear wavenumber change over a wide range. The filtered linewidth depends, amongst other factors, on the acceptance angle of the AOTF (Ward et al., 2012).

The light source of the WSI consists of a 100 W focused halogen-tungsten lamp, pinhole, collimating optics and AOTF (Gooch and Housego, AOTF model TF575-250-2-12), see Figure 5.2.

The light from the lamp is focused at 32 mm focal distance on a pinhole. The light emerged from the pinhole is collimated by a doublet lens with a focal length and diameter equal to 25 mm and 16 mm respectively.

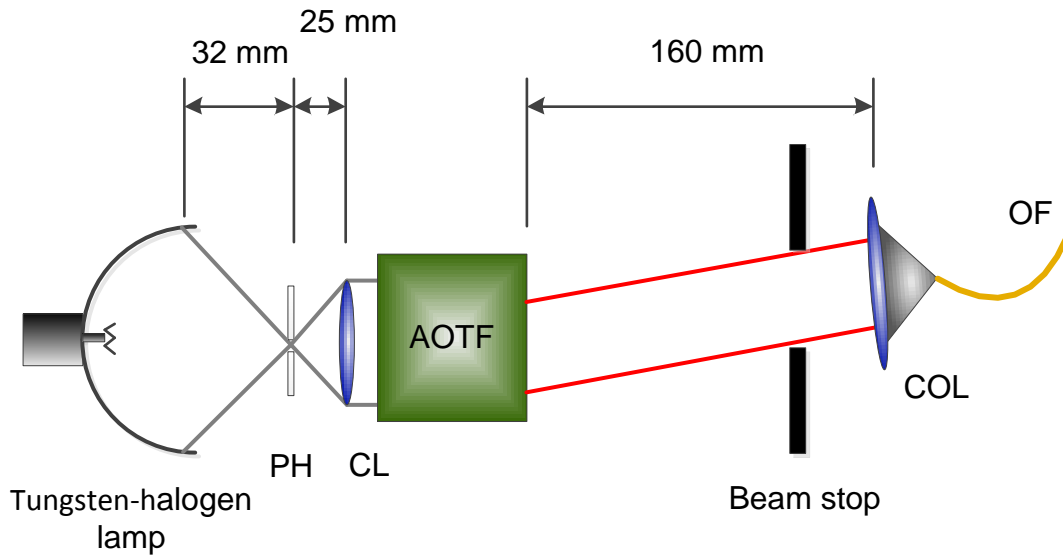


Figure 5.2 WSI light source setup

The collimation of the light is crucial to obtain the optimum diffraction efficiency from the AOTF. Collimated light divergence angle should not exceed the acceptance angle of the AOTF which is set by the manufacturer (Housego, 2010).

The divergence angle for two different pinhole diameters has been investigated. The diameter of the pinhole was first set to 1.5 mm. The collimated beam spot size was traced along 477.7 mm to determine the divergence angle. The average divergence angle at far distance was found to be greater than $\pm 1.6^\circ$. Using the same techniques, the divergence angle of a 1 mm pinhole

was found to be less than $\pm 1.2^\circ$. Since the acceptance angle of the AOTF is $\pm 1.4^\circ$, a pinhole of 1 mm diameter was used in the proposed WSI setup. Figure 5.3 shows the magnitude of the divergence angle for the collimated light at different distances.

Although reducing the pinhole diameter to less than 1 mm can produce better collimation and further decrease the divergence angle, the light intensity through the pinhole is significantly reduced which makes the fringe contrast low and hardly detectable by the CCD.

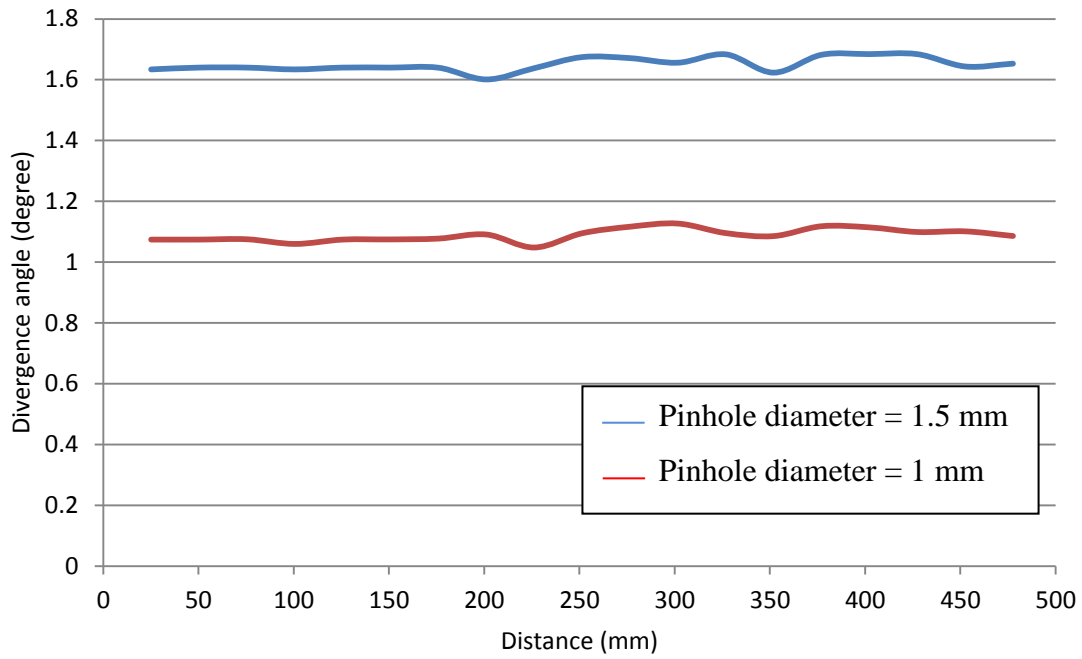


Figure 5.3 Divergence angle of the collimated white light

The AOTF is placed directly after the collimating lens. The filtered narrow band is the first order diffracted light. This order is separated from the unwanted zero order non-diffracted beam by a deflection angle. Experimentally it was found that the two beams spots are physically separated at a distance of 160 mm. A coupling lens is placed at this displacement to direct the diffracted light into a multimode optical fibre. The optical fibre transfers the light into the WSI; the zero order (non-diffracted) light is blocked by a beam stop.

5.3.1 Wavelength Scanning Characterisation

The tuning relationship between the AOTF driving frequency and filtered wavelength should be identified to ensure the linearity of the scanning process. The central wavelength of light that is diffracted from the AOTF can be determined from the equation (Xu and Stroud, 1992),

$$\lambda = \frac{\Delta n v_a}{f_a} \sqrt{\sin^4 \theta_i + \sin^2 2\theta_i} \quad (5.1)$$

where λ is the filtered wavelength, Δn is the birefringent crystal refractive index, v_a and f_a are the velocity and frequency of the propagated acoustic wave respectively and θ_i is the incident angle of the entrance light beam which is a constant.

Equation 5.1 shows that the selected wavelength depends on the acoustic wave frequency as supplied by the driver to the AOTF. Thus when the frequency is changed linearly, the wavenumber can be scanned linearly, assuming Δn is constant over the applied frequency range. As such, this scanning process can provide a linear phase shift for the interference pattern.

The scanning linearity was investigated by changing the driving frequency from 80.07 MHz to 95.70 MHz in 0.061 MHz incremental steps. This driving process was achieved using AOTF frequency driver model 63030-200-2AMDFS-A from Gooch and Housego (see Appendix A.1). The AOTF wavelength spectrum was measured by a grating spectrometer (Thorlabs, model CCS100), having spectral resolution of 1 nm (FWHM). This spectrometer is connected to the end of the optical fibre shown in Figure 5.2.

For the scanned frequency range, the wavelength was tuned from 683.42 nm to 590.98 nm as shown in Figure 5.4. The wavenumber is shifted linearly with a maximum error equal to 0.17%. This error is approximately equivalent to 1.1 nm which is close to the ccs100 spectrometer resolution. This error could be caused by either a drift in the driving frequency supplied to the AOTF or a measurement error in the wavelength spectrum due to the resolution limitation of the spectrometer.

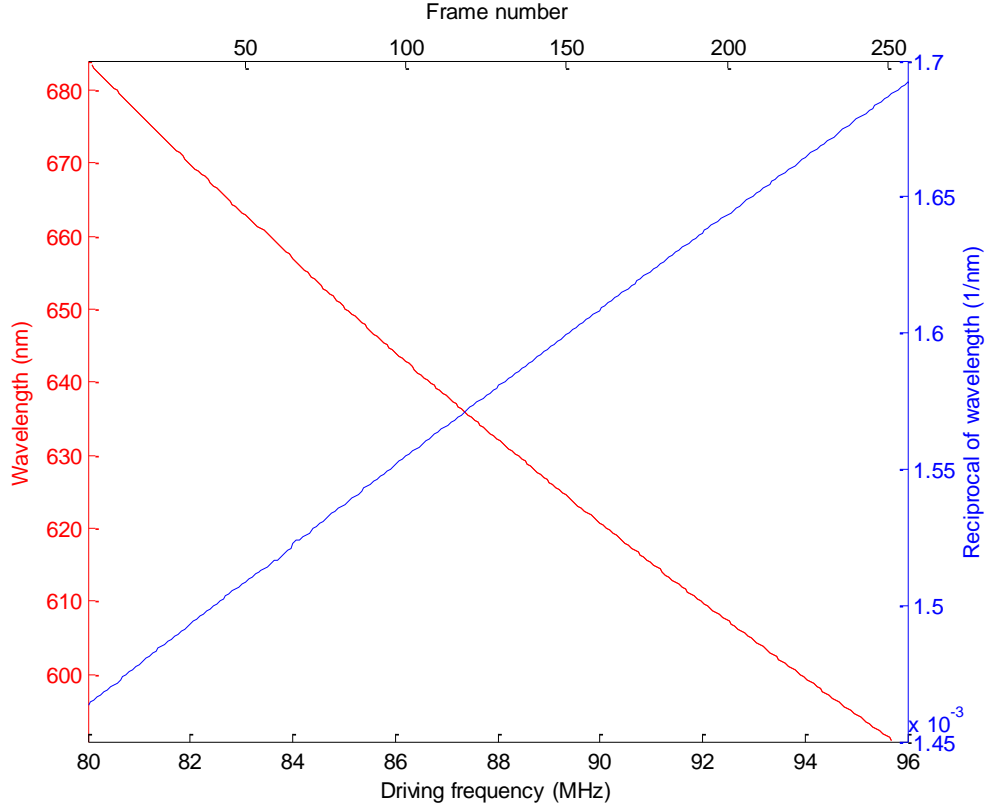


Figure 5.4 Wavelength scanning range versus driving frequency

The linewidth of the filtered wavelength λ_{FWHM} can be expressed by the following equation (Ewa G. Bucher, 1999)

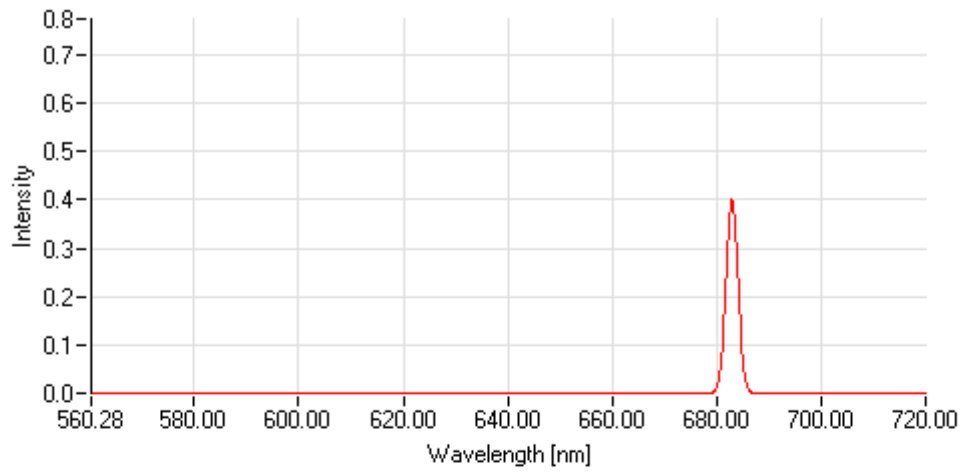
$$\lambda_{FWHM} = \frac{0.9\lambda^2}{\Delta n L \sin^2 \theta_i} \quad (5.2)$$

where L is the interaction length between the acoustic wave and the optical radiation. If all other terms are assumed to be constant, λ_{FWHM} is proportional to the square of the wavelength.

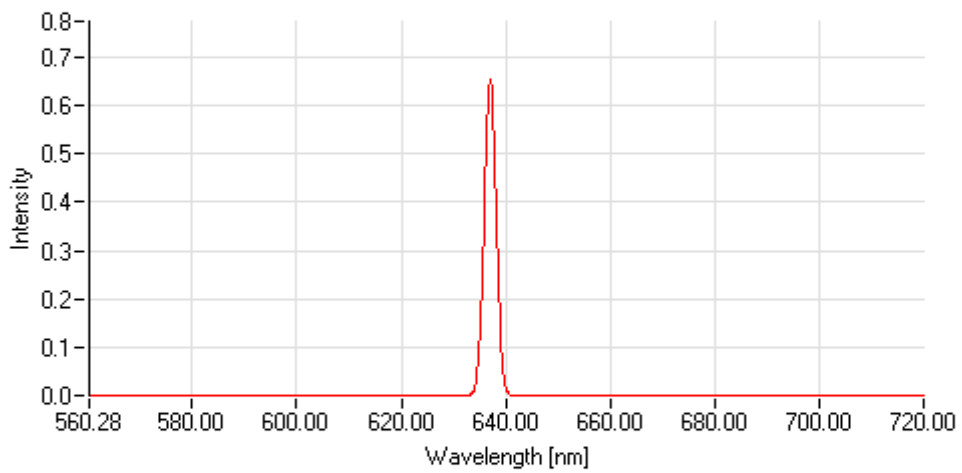
The λ_{FWHM} was evaluated at three different scan positions using the CCS100 spectrometer, results are shown in Figure 5.5. The applied radio frequency power was fixed at 0.5 W. It was found that the linewidth of the wavelength at the upper end of the scanning range ($\lambda = 683.42$ nm) was equal to 2.2 nm. The linewidth narrows at shorter wavelengths however: 2 nm at $\lambda = 637.39$ nm and

1.7 nm at $\lambda=590.98$ nm. These measurements illustrate that better linewidth resolution occurs at shorter wavelengths which is in agreement with Equation 5.2.

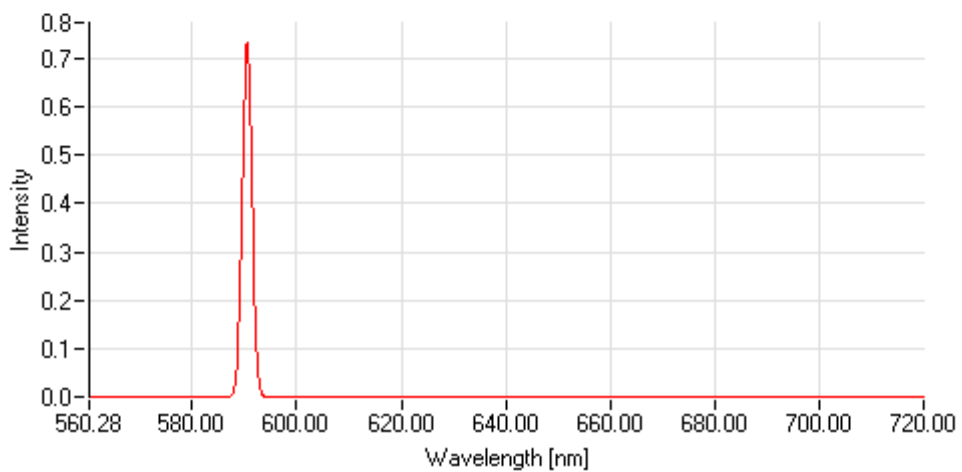
Narrowing the linewidth could significantly reduce the light intensity available for detection by the CCD. It was found that the intensity light throughput is insufficient when the filtered wavelength is scanned beyond a lower limit of 590.98 nm. Scanning above an upper limit of 683.42 nm resulted in the linewidth exceeding 2.2 nm. Consequently this leads to a reduced the coherence length of the source light. Therefore, the scan range was fixed from 683.42 - 590.98 nm.



(a)



(b)



(c)

Figure 5.5 Wavelength bandwidth (a) linewidth at 683.42 nm = 2.2 nm, (b) linewidth at 637.39 nm = 2 nm, and (c) linewidth at 590.98 nm = 1.7nm

5.4 WSI Experimental Setup

The optical configuration of the WSI system is based on a Linnik interferometer illuminated by the tunable light source (see Figure 5.1). The filtered wavelength from the tunable light system is guided to the interferometer by a multimode optical fibre having a 600 μm core diameter. The benefit of using an optical fibre is to physically separate the light source from the interferometer so that the source of illumination can be placed away from the measurement area. This feature can provide a simple measurement head which could be fixed on a manufacturing machine. The output beam of the optical fibre is collimated by a doublet lens of 30 mm focal length and coupled to the interferometer.

The interferometer is composed from two balanced optical arms split by a beam splitter (B.S). The two arms are noted as the reference arm that ends with a reference mirror and the measurement arm that ends with a measurement sample. Additionally, two matched microscope objectives lenses (type M Plan Apo) are placed at 44.7 mm distance from the B.S and aligned in the optical path of each arm.

The collimated beam that emerges from the fibre is split by a beam splitter into two halves which then propagate along the two arms. The reflected beams from the measurement sample and the reference mirror are recombined by the B.S to produce interference fringes.

The interference fringes are focused into a two dimensional CCD array by a doublet imaging lens of 250 mm focal length. The CCD camera (JoinHope Image, model OK-AM1131) is connected to a frame grabber that stores and transfers the captured images to the CPU memory of the computer.

A National Instruments data acquisition card (DAQ) and pulse to binary converter circuit (PBC) are used to supply a binary word to the AOTF frequency driver in order to select the frequency of the acoustic wave (see Appendix A.1). Figure 5.6 shows the experimental WSI setup.

The items that are labelled SLED, edge-pass filter, photo-detector and PI controller are related to the stabilisation and will be described in Chapter 8. This system has been prototyped as shown in Appendix A.2.

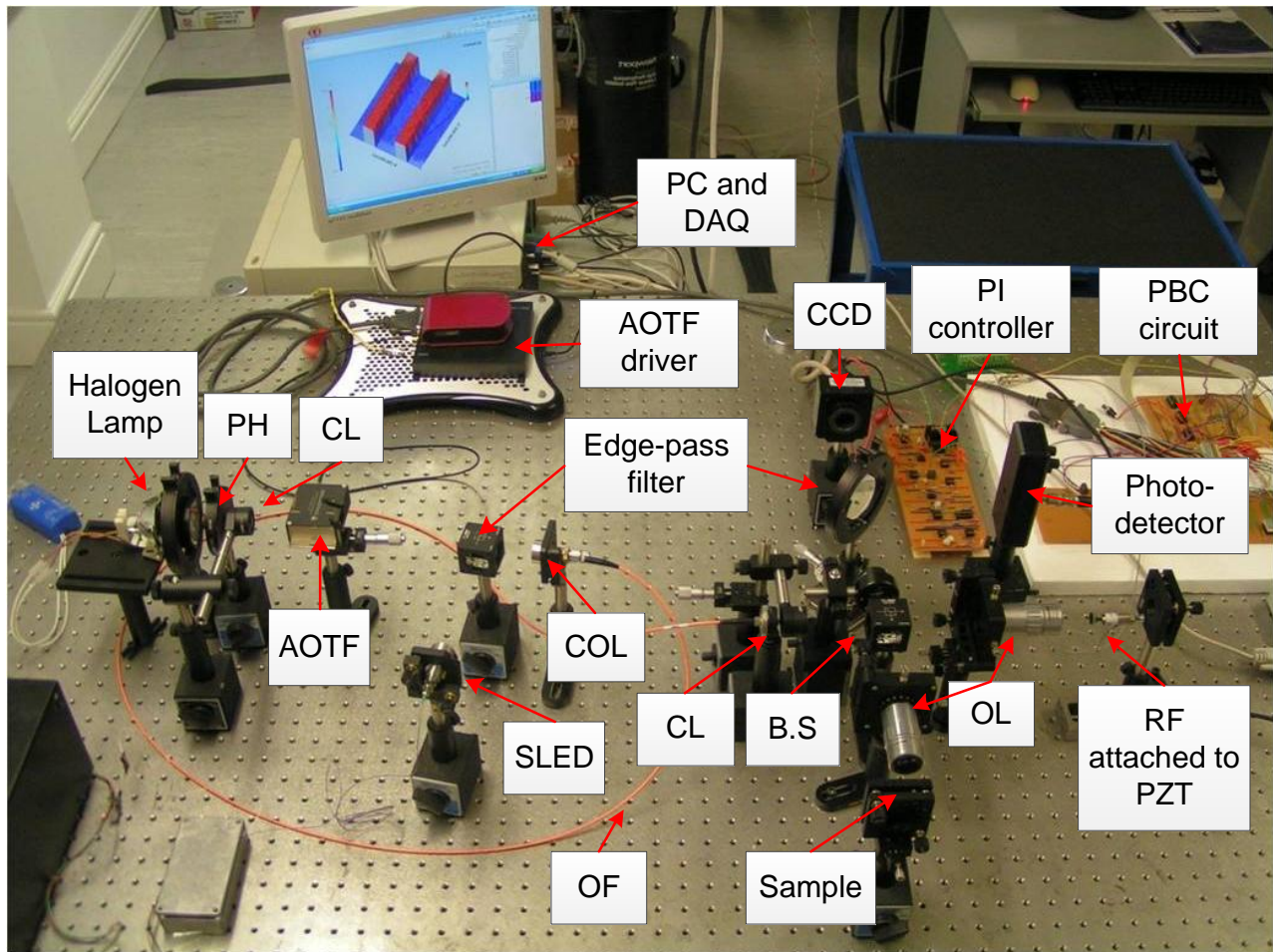


Figure 5.6 The WSI experimental setup, where PH: Pinhole, CL: Collimating lens, OF: Optical fibre, COL: Coupling lens, B.S: Beam splitter, OL: Objective lens, RF: Reference mirror, FG: Frame grabber, PBC: Pulse to binary converter

5.5 The WSI Operation and Data Capturing

The prototype setup as described was integrated with a multi-axis stage purchased from Aerotech Inc. (see Figure 5.7). The resolution of the rotary stage can be as high as 3.2 arcsec while

the linear stage is up to $\pm 6 \mu\text{m}$ (Aerotech, 2010a, Aerotech, 2010b). This integration facilitates the instrument alignment with samples especially those with large dimensions.

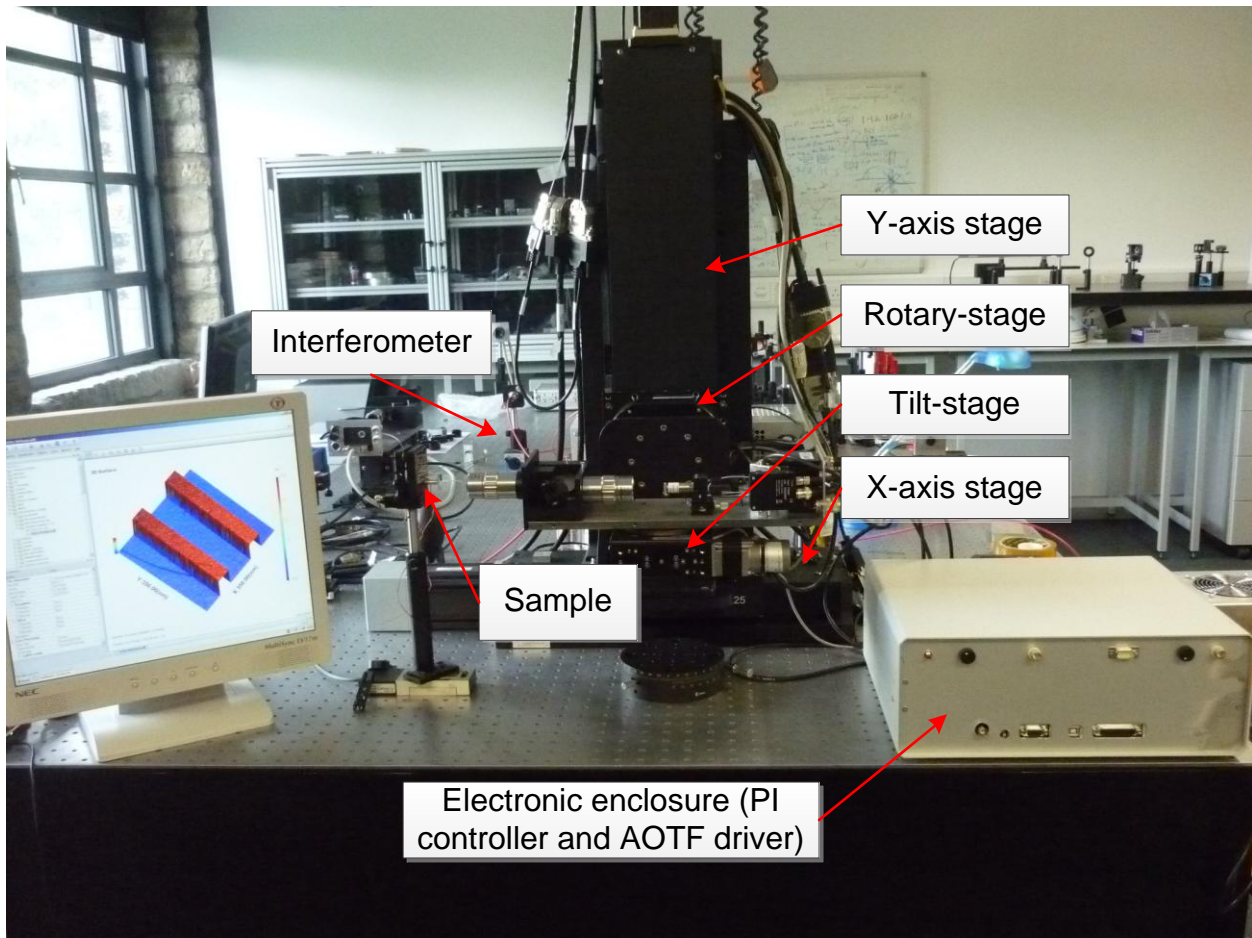


Figure 5.7 WSI system

A sample was placed at the end of the interferometer measurement arm. The system was first operated at a specific wavelength to align the sample and obtain an interference fringes. This was achieved by driving the AOTF at a constant frequency (e.g. 80.07 MHz). The sample was aligned within the focal depth of the objective lens. Since the tuned light has coherence length exceeding the focal depth, interference fringe for the sample were obtained and captured by the CCD. The interference was composed of a single fringe pattern spread over the entire sample area, since the sample plane is aligned to be coincident with the virtual reference mirror plane.

After the alignment and interference fringe detection, the WSI measurement operation was performed by scanning the wavelength from 683.42 nm to 590.98 nm in approximately 0.36 nm decrements. Simultaneously, at each scanned step a single frame is captured with 640 x 480 pixels; a total of 256 frames are needed to cover the entire scanning range. The captured frames are shown in Figure 5.8.

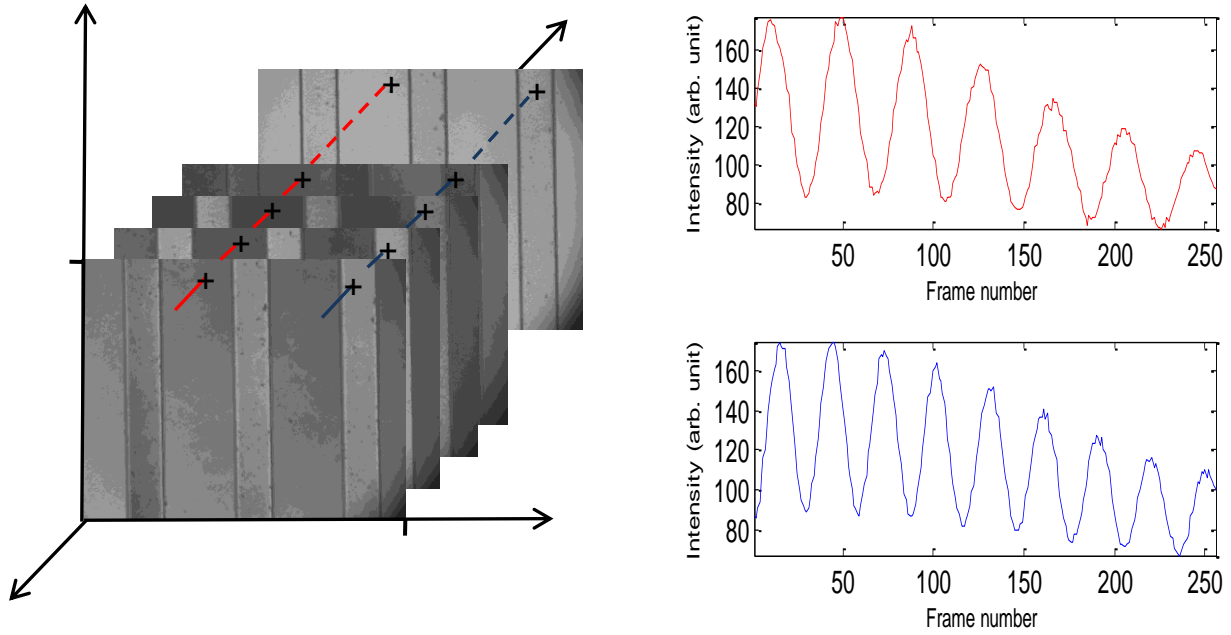


Figure 5.8 Captured frames by the CCD

Each captured frame has a unique wavelength with each frame pixel representing a specific point on the measured surface. The interference pattern obtained from a single pixel across all captured frames is subjected to the interference equation:

$$I_{xy}(\lambda_i) = a_{xy}(\lambda_i) + b_{xy}(\lambda_i) \cos(4\pi k_i h_{xy}) \quad (5.3)$$

where λ_i is the tuned wavelength, x and y indices are the pixel number in the horizontal and vertical directions respectively of the CCD detector, a is the interference DC bias, b is amplitude of interference contrast variation, k is the wavenumber and h is the path difference between corresponding point on the reference plane and sample. By tuning the wavenumber in a linear

manner, a periodic sinusoidal intensity distribution is obtained. The sinusoidal patterns in Figure 5.8 show the interference signals of two points upon the sample (red and blue traces). It can be seen that the two patterns having difference frequencies and number of peaks and valleys (i.e. difference phase shifts at a fixed wavelength scanning range). This indicates that the two points having different optical path lengths. Chapter 6 introduces and describes four algorithms to analyse the interference pattern and determine the surface height of the sample.

One of the main advantages of this system is the measurement speed which is limited only by the CCD capturing rate. This is because the AOTF has a fast scanning rate over a wide range. The wavelength scanning step can be switched within $0.2\ \mu\text{s}$ whereas the fastest capturing rate is several milliseconds (Housego, 2011, Andor-Technology, 2012). Typically 256 frames are acquired to analyse the surface height yielding a measurement speed of approximately 2.6 s since the CCD used has speed rate of 100 frames/second.

5.6 Intensity Behaviour of AOTF Filtered light

The intensity of filtered light diffracted from the AOTF has been investigated over the entire wavelength scanning range using the WSI system with the measurement arm blocked so that no interference fringes are created. The light reflected back from the measurement arm of the interferometer was blocked and the beam reflected from the reference arm was recorded by a single CCD pixel.

The CCD pixel recorded 256 intensity values during the scanning process. The wavelength was scanned from 683.42 nm to 590.98 nm. Figure 5.9 shows the recorded intensity values without interference effects. This figure shows a slow decay envelope function toward shorter wavelengths. This effect is identified in this thesis as a background intensity variation.

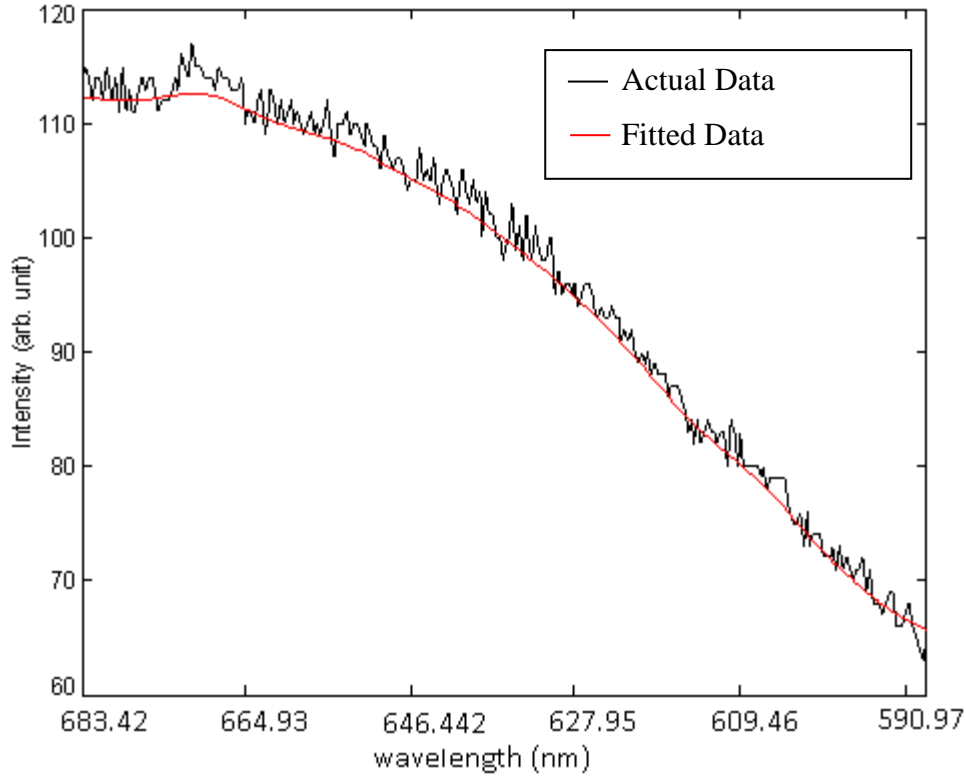


Figure 5.9 Background intensity variation during wavelength scanning

The main reason for this decay could be the narrowing of the spectral linewidth at shorter wavelengths. This narrowing decreases the light intensity and effectively reduces the number of photons captured by the CCD pixel.

The optical components, such as collimating and objective lenses, are corrected for the visible region. Furthermore, they are anti-reflection coated to reduce the reflectance to 0.3%. Both the CCD response and the tungsten-halogen light source emission spectrum increases toward shorter wavelengths which is inconsistent with Figure 5.9 (see Appendix A.3). Therefore, these aspects are ignored and taken to have no relation with background intensity variation.

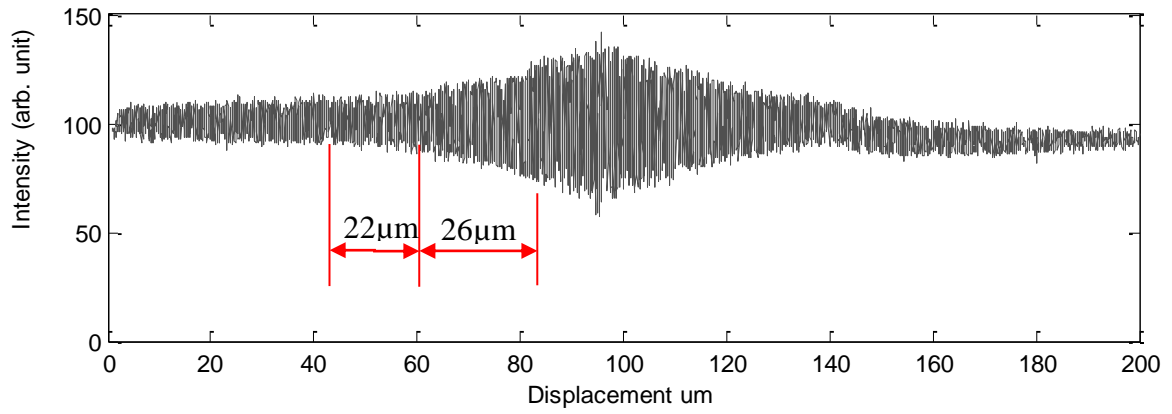
5.7 Coherence length of WSI

The proposed WSI system has a limited coherence length since the spectral line width has a moderate resolution of approximately 2 nm. The fringe visibility is a function of the OPD. As the

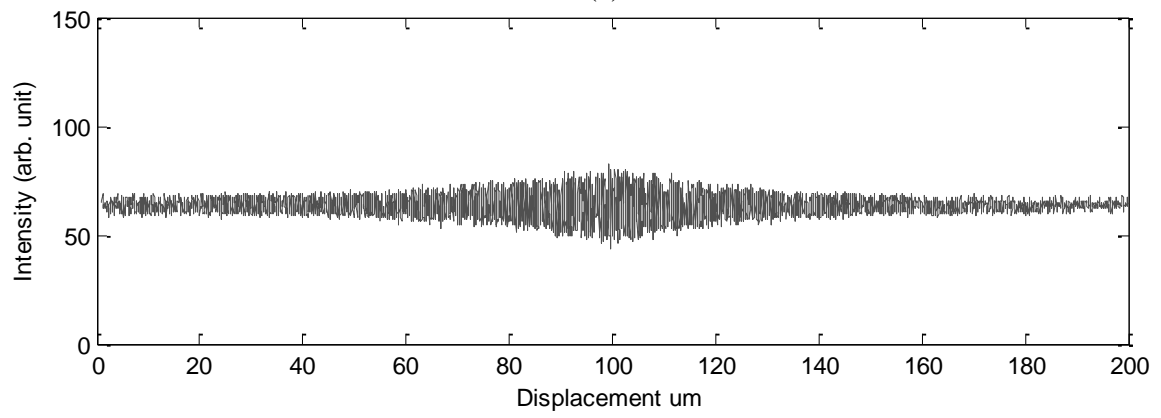
OPD increases, the visibility of the fringes decreased and vanishes if the OPD exceeds the coherence length. As such, the measurement range is related directly to coherence length.

The coherence length can be measured by monitoring the fringe visibility when the OPD is changed in the WSI. To investigate the coherence length over the entire scanning range, three wavelengths were examined individually; the initial (683.42 nm), middle (637.39 nm), and final (590.98 nm) wavelengths. At each wavelength, the reference mirror was moved 200 μm by an accurate stepper motor around the zero OPD. Simultaneously, an interferogram produced from a single point was detected by a CCD single pixel. The experiment was been carried out using a 2x magnification objective lens.

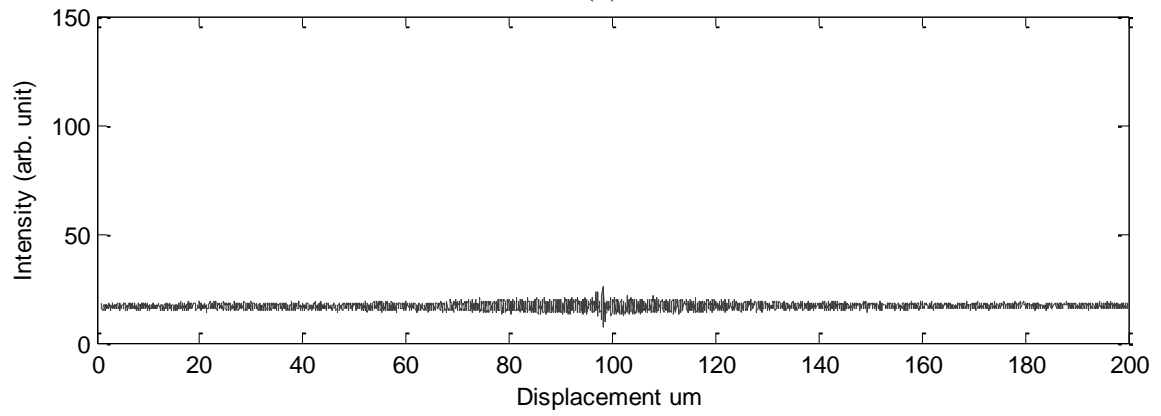
Figure 5.10 shows that the FWHM of the coherence function occurs at approximately ± 26 μm and the $1/e^2$ point at approximately ± 48 μm . Although the intensity level is varied at different wavelengths, the ratio of fringe visibility variation is almost the same for the entire scanning range as shown in Figure 5.11. Obviously, the change in the intensity level between the three wavelengths is due to background intensity variation as discussed in Section 5.6. Therefore, if the coherence length is taken to be the $1/e^2$ value of the coherence function, the measurement range is ± 48 μm . This result is consistent with the theoretical coherence length calculation that is stated in Table 3.1.



(a)



(b)



(c)

Figure 5.10 Coherence length of WSI at (a) $\lambda=683.42$ nm (b) $\lambda=637.39$ nm (c) $\lambda=590.98$ nm

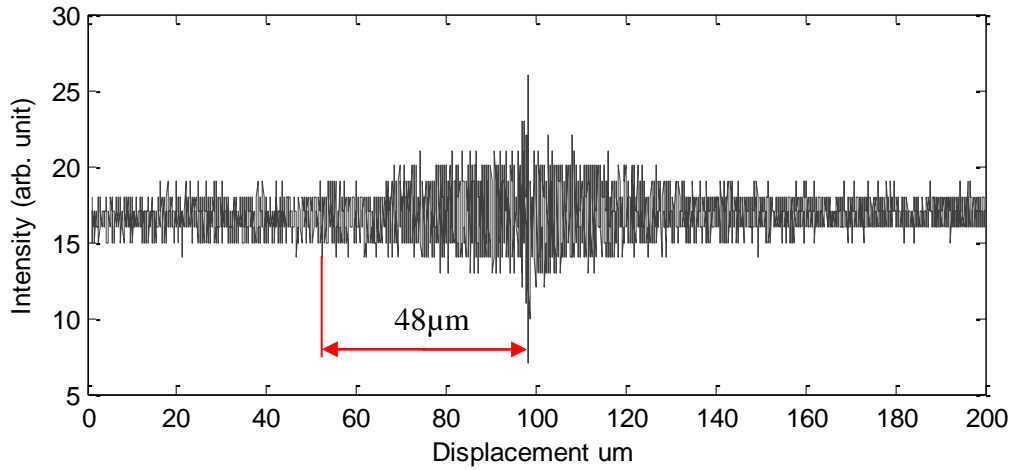


Figure 5.11 Magnified scale for coherence length of WSI at $\lambda=590.98$ nm

5.8 Lateral Range and Resolution

The lateral range and resolution vary for different objective lenses. The lateral range or field of view (FOV) is defined as the area of sample observable by the instrument, while the lateral resolution is defined as the minimum separation between two points that can be distinguished by the instrument.

The lateral range depends mainly on the light spot size on the sample and the number of the CCD pixels (P_n). The lateral range has been determined by using a stage micrometer with 50 μm division. In one example, the stage micrometer was captured with 2X objective lens and a 640x480 pixel CCD array as shown in Figure 5.12. The contrast of the red line shown in Figure 5.12 was examined and the results shown in Figure 5.13.

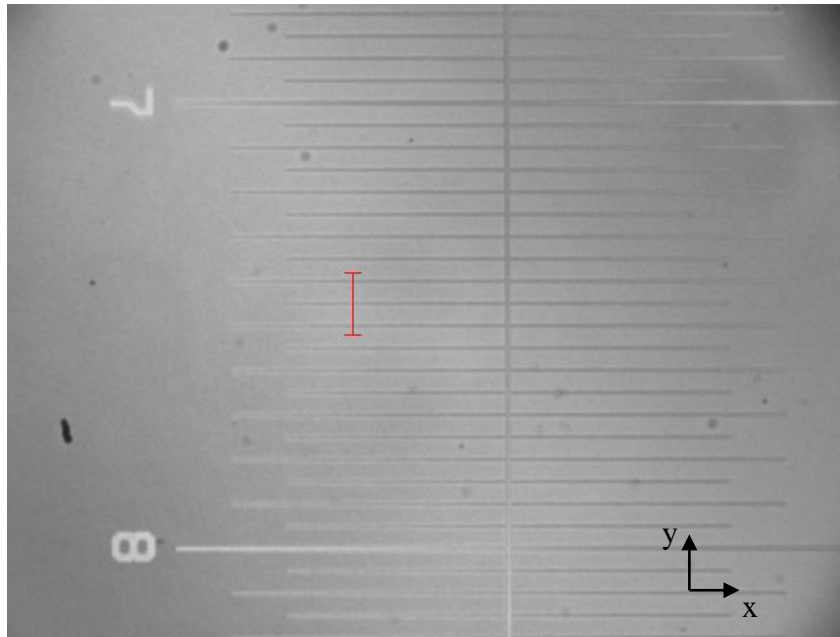


Figure 5.12 Captured frame for micrometer stage with 50µm divisions using a 2X lens

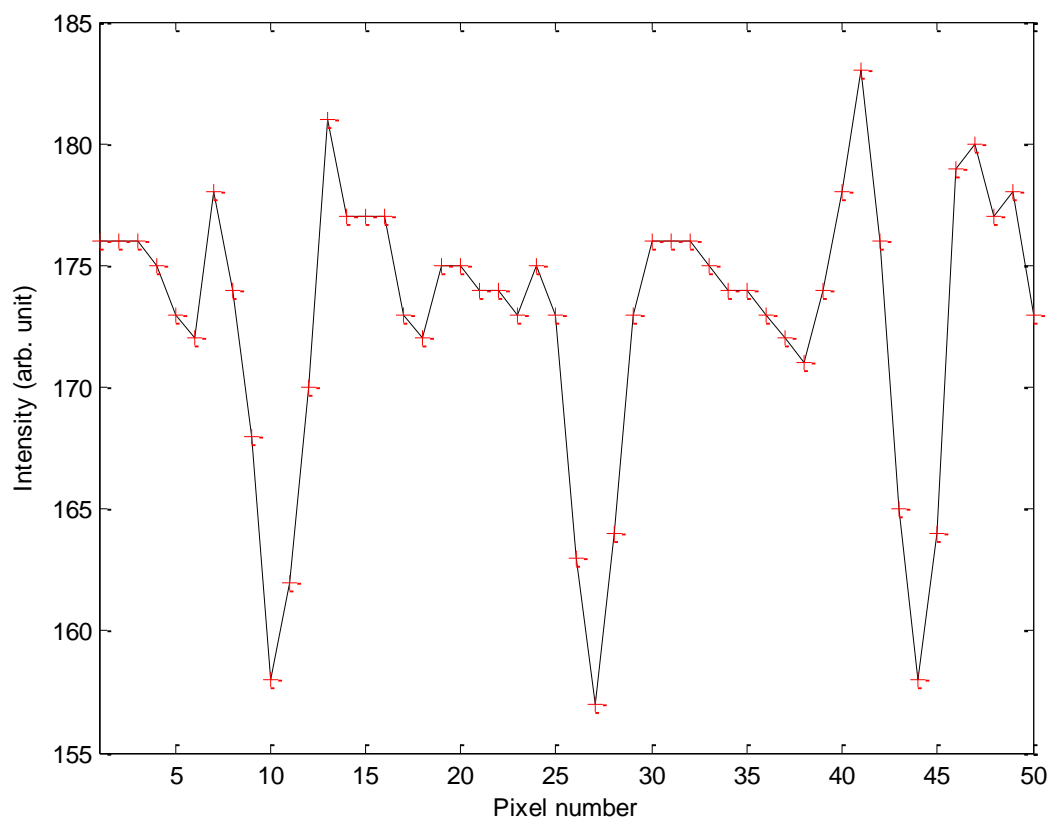


Figure 5.13 Relationship between micrometer stage and pixelation for 2X lens

It is found that every 17 pixels covers 50 μm of the imaged surface. As such, each pixel division (P_d) in the vertical direction represents 2.941 μm on the sample. Because of the cylindrical symmetry of the objective lenses, the horizontal pixel division is assumed to be equal to the pixel division in vertical direction. Thus the lateral range in both x and y directions (Lr_x, Lr_y) can be calculated using:

$$Lr = P_d * P_n \quad (5.3)$$

$$Lr_x = 2.941 \mu\text{m} * 640 = 1.882 \text{ mm}$$

$$Lr_y = 2.941 \mu\text{m} * 480 = 1.412 \text{ mm}$$

The same concept is applied to a 5X objective lens. It can be seen from Table 5.1 that the lateral range is reduced when the objective magnification increases.

Table 5.1 Lateral ranges for different objective lenses

Objective lens	Pixel division (μm)	Lateral range (mm)
2X	2.941	1.882 x 1.412
5X	1.191	0.762 x 0.572

Practically, the lateral resolution (r) depends on the contrast variation obtained from the Airy disk patterns of two close points (Xu and Stroud, 1992). If the contrast variation of the two points is within a specific resolution criterion such as Rayleigh (contrast variation between two points is 26.4%), the displacement between them represents the lateral resolution (Spring et al., 2009).

The Rayleigh criterion is used to estimate the lateral resolution. For the optimum situation where the aberration is ignored, the signal to noise ratio is considered to be sufficient, and the number of pixels meet with the sampling requirement, the lateral resolution can be expressed as:

$$r = 0.61 \frac{\lambda}{NA} \quad (5.4)$$

where r is the lateral resolution, NA is the numerical aperture of the lens and λ is the wavelength of the source light. Since the proposed system is based on a wavelength scanning technique, the lateral resolution is not constant. However, the lateral resolution is estimated at the highest wavelength (worst case) which is 683.42 nm. Using equation 5.4, the lateral resolution was calculated for different objective lenses and the results are shown in Table 5.2.

Table 5.2 The lateral resolution based on Rayleigh criterion

Objective lens	Numerical aperture (NA)	Lateral resolution (μm)
2X	0.055	7.58
5X	0.14	2.98

5.9 Summary

This chapter has introduced WSI with an electronically tuned AOTF and a high speed CCD camera. The AOTF provides a fast wavelength scanning technique to produce a linear phase shift and as such, the overall acquisition time is limited by the CCD speed only.

The best scanning range for the described system was found to be from 683.42 nm to 590.98 nm. The spectral linewidth of the filtered light for this range varied from 2.2 nm to 1.7 nm. This change in linewidth produced a drop in the background intensity when the filtered wavelength was tuned from longer to shorter wavelengths.

The scanning process produces a periodic sinusoidal interference pattern from each CCD pixel which contains the surface height information. By considering the all the pixels in the CCD array areal measurement can be achieved.

The coherence length of the filtered light was found to be $\pm 48 \mu\text{m}$. Consequently, the measurement range of the described instrument is limited to this value. Nevertheless, the depth of focus of commercial microscope objectives, which are used for surface inspection, is less than the

filtered light coherence length. For instance, the depth of focus for 2X microscope objective is approximately 92 μm (Mitutoyo, 2010).

The lateral range and resolution of the described system is dependent on the microscope objective lens. For instance, the lateral range for 2X lens is 1.882 x 1.412 mm with a resolution of approximately 7.58 μm .

The ability of WSI to measure with no mechanical scanning makes the system suitable to be integrated with machine tools for on-line/in-process part inspection, especially if a stabilisation feature can be added to this instrument, as will be discussed in Chapter 8.

6. Fringe Analysis and Algorithm

6.1 Introduction

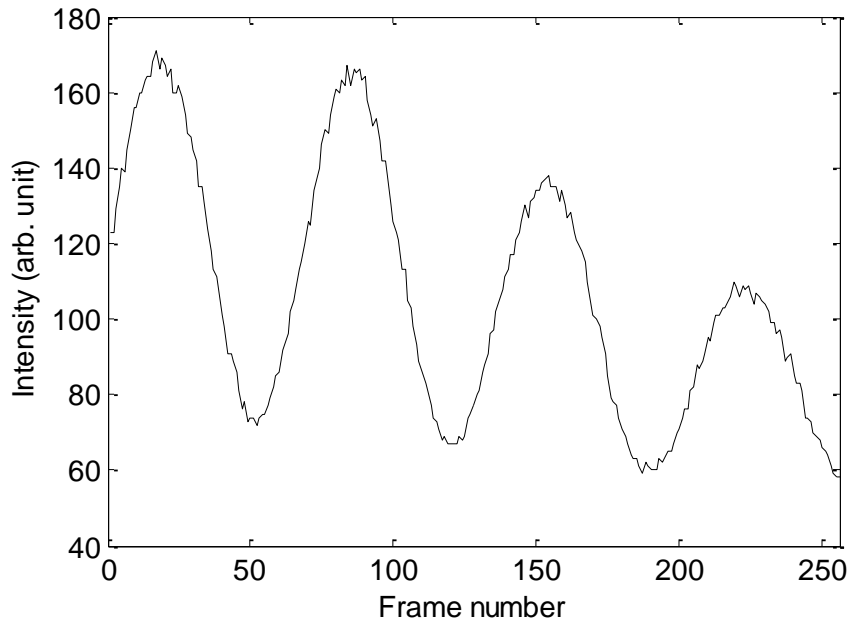
The measurement and retrieval of the surface topography information from the periodic interference fringes obtained with the WSI can be achieved by using a variety of algorithms. The resolution and accuracy of the measurements depend on the computing resolution of the algorithm used. This chapter discusses four different algorithms used to evaluate the interference patterns obtained. The mathematical description of these algorithms and their computing resolution capabilities are presented. Two step height samples and a V-groove sample are measured using the WSI and analysed using each of the four algorithms. Using specific objective lenses, a fixed wavelength scanning range and certain number of captured frames, the experimental results demonstrate that the determination of absolute optical path difference yields the profile of a surface with a resolution which varied from 11 nm to several micrometres depending on the algorithm used to analyse the captured data.

6.2 Interference Pattern of the Fringe

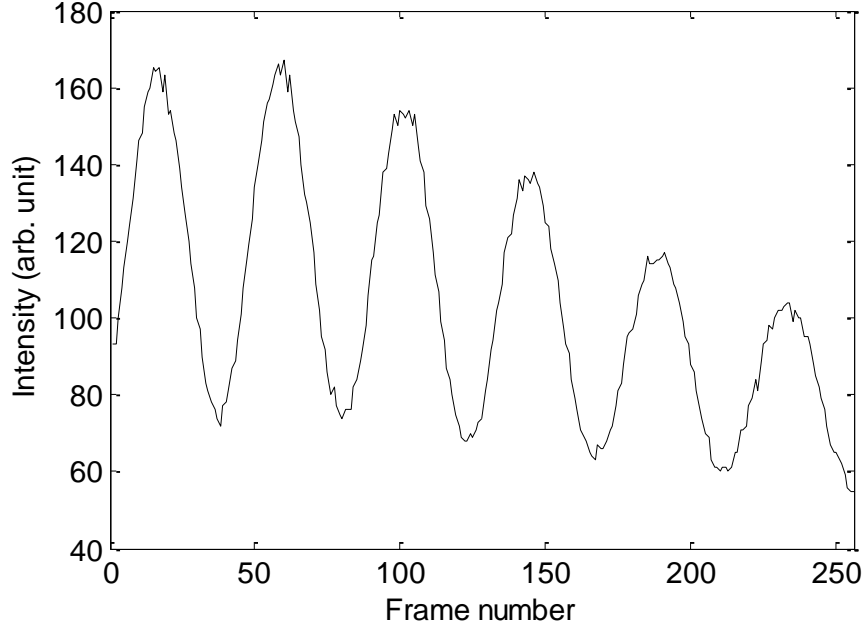
The WSI measurement system described in Chapter 5 was used to measure five structured samples. The samples are three different step height standards, a V groove and a Sine wave. The step height standards have values of 100 nm, 4.707 μm and 9.758 μm . The V groove sample has a value of a side wall length equal to 23.2 μm and a slope angle equal to 45° . The Sine wave sample has an amplitude of 0.61 μm and a wave period of 14 μm . The measurement results of the samples of 9.758 μm step height and Sine wave are demonstrated in Appendix B.

The wavelength scanning process was achieved by using an AOTF to filter a narrow band from a tungsten-halogen light source. The filtered wavelength was scanned in a range from 683.42 - 590.98 nm.

The obtained spectral linewidth of the filtered light is dependent on the tuned wavelength and ranges from 1.7 to 2 nm at FWHM. This linewidth range can provide a coherence length of up to 96 μm . The filtered wavelength was scanned in approximately 0.36 nm increments. At each incremental step, a single frame was captured with 640 x 480 pixels. A total of 256 frames were captured throughout the entire scanned wavelength range. The overall capture time was approximately 2.6 seconds. Each captured frame has a unique wavelength with each frame pixel representing a specific point on the measured surface. By considering a single pixel across all the captured frames, a periodic sinusoidal intensity distribution is obtained for a single point on the measured surface (see Figure 6.1). The frequency of the intensity pattern is proportionally related to the OPD. Figure 6.1 shows two interference patterns, each at a different point on the sample. The absolute distance difference between the two points is approximately 4.707 μm where the point in Figure 6.1a was closer to the virtual reference plane. It can be seen from this figure that the frequency and number of peaks and valleys of the pattern are different within the same scanned wavelength range.



(a)



(b)

Figure 6.1 Interference patterns of two points on the sample

Equation 6.1 is a mathematical representation of the intensity distribution (Hariharan, 1991).

$$I_{xy}(\lambda) = a_{xy}(\lambda) + b_{xy}(\lambda) \cos(\phi_{xy}(\lambda)) \quad (6.1)$$

where $a_{xy}(\lambda)$ and $b_{xy}(\lambda)$ are interference intensity bias and amplitude of the fringe contrast respectively, and λ is the scanned wavelength. The x and y indices are the pixel number in the horizontal and vertical directions respectively of the CCD detector.

The phase shift of the interference pattern due to the wavelength shift, $\Delta\phi_{xy}(\lambda)$ is given by,

$$\Delta\phi_{xy}(\lambda) = \frac{4\pi}{\lambda_s} h_{xy} \quad (6.2)$$

where h_{xy} is the height of the sample pixel and the λ_s is the synthetic wavelength which is defined by,

$$\lambda_s = \frac{\lambda_{\max} \lambda_{\min}}{\lambda_{\max} - \lambda_{\min}} \quad (6.3)$$

The synthetic wavelength for the given range is $\lambda_s = 4.37 \mu\text{m}$. The height, h_{xy} is related to half of the total optical path difference in the interferometer and can therefore be obtained by determining the absolute of the phase change through the entire wavelength scanning range as shown in Equation 6.4.

$$h_{xy} = \frac{\Delta\phi_{xy} \lambda_s}{4\pi} \quad (6.4)$$

Various algorithms may be used to determine the phase change relating to the surface height of the measured surface. Since the obtained interference patterns have a periodic sinusoidal form, the most common methods to analyse these patterns are Fourier transform and detection of peak positions. The following section describes the resolution of the measurement using four algorithms based on using FFT and localising the peaks of the fringes.

6.3 Analysis of Phase Determining Algorithms

The resolution of the surface height measurement using a WSI varies depending on the algorithms used to evaluate the phase shift obtained from the wavelength scanning process. The more accurate the determination of the phase shift, the higher the resolution that can be obtained. There have been several computing algorithms proposed previously to determine the phase shift (Yamamoto et al., 2001, Takeda et al., 1982, Schwider and Zhou, 1994, Snyder, 1980).

This chapter discusses four algorithms and uses them to analyse real measurement data acquired from the WSI described in Chapter 5. The first algorithm uses the power spectral density of the interference pattern to determine the phase shift by identifying the fundamental frequency. The spectrum is obtained by using simple FFT (see Appendix B.1.1). The second algorithm uses a fitted spectrum to estimate the fundamental frequency more accurately, hence increasing the resolution (see Appendix B.1.2). The third algorithm uses FFT and inverse FFT to analyse the sinusoidal interference pattern and isolate the phase information from other unwanted data such as

the DC component and the conjugate term of the phase (see Appendix B.1.3). Finally, the fourth algorithm identifies the peak positions of the sinusoidal interference pattern with respect to the scanned wavelength range (see Appendix B.1.4).

6.3.1 Phase Shift Determination Using Simple FFT: Algorithm A

Phase shift determination using this method is based on determining the number of cycles of the sinusoidal interference pattern. The number of cycles is determined directly from identifying the Fourier peak of the spectrum as shown in Figure 6.2.

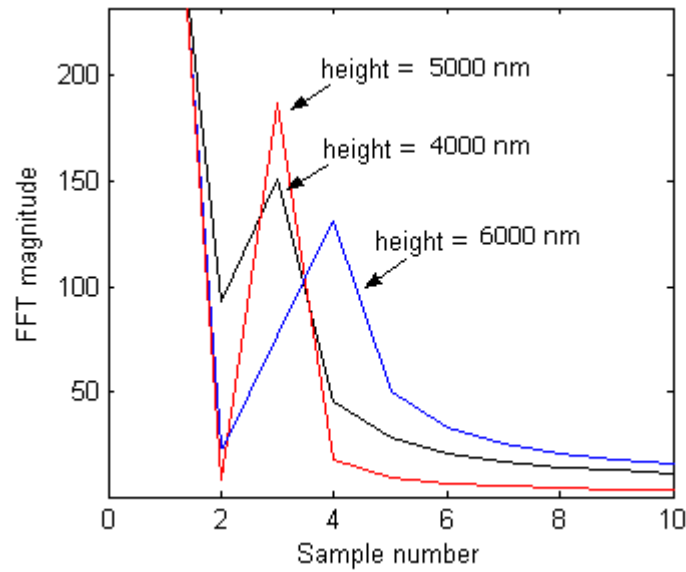


Figure 6.2 Simulated Fourier power spectral densities for different heights

Since each cycle represents a 2π phase shift, the overall phase shift can be calculated by multiplying the number of cycles by 2π . The surface height may then be calculated using Equation 6.4. The main drawback of this method is that the determination of phase shift is limited to a 2π phase interval, which limits the surface height resolution to the micron scale. The resolution can be calculated by:

$$\delta h = \frac{\delta \phi \lambda_s}{4\pi} \quad (6.5)$$

where δh and $\delta \varphi$ are the resolutions of the surface height and phase respectively. $\delta \varphi$ is equal to 2π for this algorithm. Substituting into Equation 6.5 yields:

$$\delta h = \frac{\lambda_s}{2} \quad (6.6)$$

Thus the height resolution for the given synthetic wavelength ($\lambda_s=4.37\mu\text{m}$) in this research is $2.18\mu\text{m}$.

6.3.2 Phase Shift determination Using Fitted FFT: Algorithm B

Yamamoto et al. (2001) used a fitted spectrum to determine the phase shift. The surface height resolution is improved to sub-micrometre by using parabolic fitting onto three data points around the Fourier peak. The general formula of a parabola is given by the equation:

$$f(x) = Ax^2 + Bx + C \quad (6.7)$$

where A , B and C are the parameters of the quadratic equation. The least squares quadratic polynomial approximation and Gaussian elimination can be used to fit the data (see Figure 6.3) and determine the values of A , B and C . Since the slope at the peak of the curve is equal to zero, the Fourier peak position can be calculated from the first derivative of Equation 6.7 as:

$$x = -B / (2A) \quad (6.8)$$

This method is sensitive enough to detect phase variations of less than 2π . The estimated Fourier peak position is multiplied by 2π and is taken as the value of the phase shift in Equation 6.4 in order to calculate the surface height.

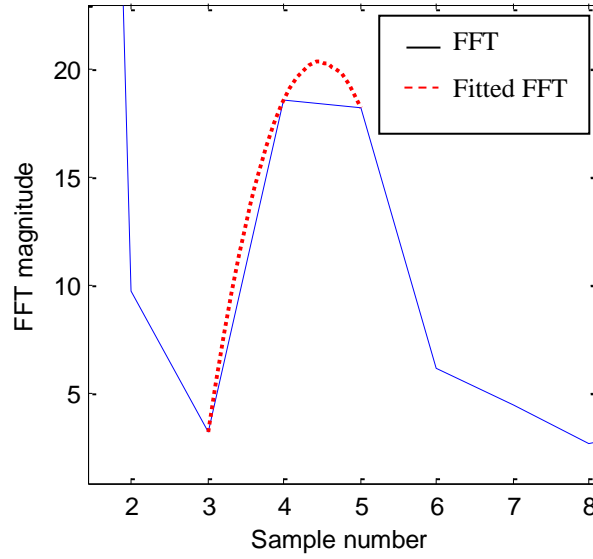


Figure 6.3 Simulated Fourier power spectral density for 7490 nm single point height

6.3.3 Interference Pattern Analysis Using Fourier Transform: Algorithm C

The resolution of the WSI can be much improved by determining the optical path difference from the slope of the phase shift. This method was first introduced by Takeda et.al in 1982 using a wedge type interferometer. In contrast to the wedge interferometer, which acquires the interference fringe pattern across the measured surface profile, the WSI captures the pattern from a single point. As such, the entire phase shift values are related to a unique surface height; only one pixel is needed to acquire the pattern. By analysing all the captured pixels, areal topography can be obtained.

Since the wavenumber is scanned linearly, the phase shift also changes linearly. Consequently the rate of change of the phase with wavenumber can be used to determine the optical path difference, and thus height, as stated in Equation 6.4. An example of an interference intensity pattern of one pixel obtained from the wavelength scanning process is shown in Figure 6.1.

The sinusoidal pattern in Figure 6.1 has a slowly changing envelope function. This envelope is due in part to the changing width of the AOTF filter characteristic where this change varies the light throughput over the tuned wavelength range. This systematic variation is eliminated by dividing the interference intensity pattern over the background intensity distribution along the

scanned wavelength range, as explained in Chapter 5 (see Figure 5.9). The compensated intensity pattern resulting from this operation is shown in Figure 6.4.

Equation 6.1, which gives a mathematical expression for Figure 6.4, can be presented in more convenient form as:

$$I_{xy}(i) = a_{xy}(i) + \frac{1}{2}b_{xy} e^{j\phi(i)} - \frac{1}{2}b_{xy} e^{-j\phi(i)} \quad (6.9)$$

Equation 6.9 may be simplified by using the notation, $c = \frac{1}{2}b e^{j\phi(i)}$ and $c^* = \frac{1}{2}b e^{-j\phi(i)}$

leading to,

$$I_{xy}(i) = a_{xy} + c_{xy} + c_{xy}^* \quad (6.10)$$

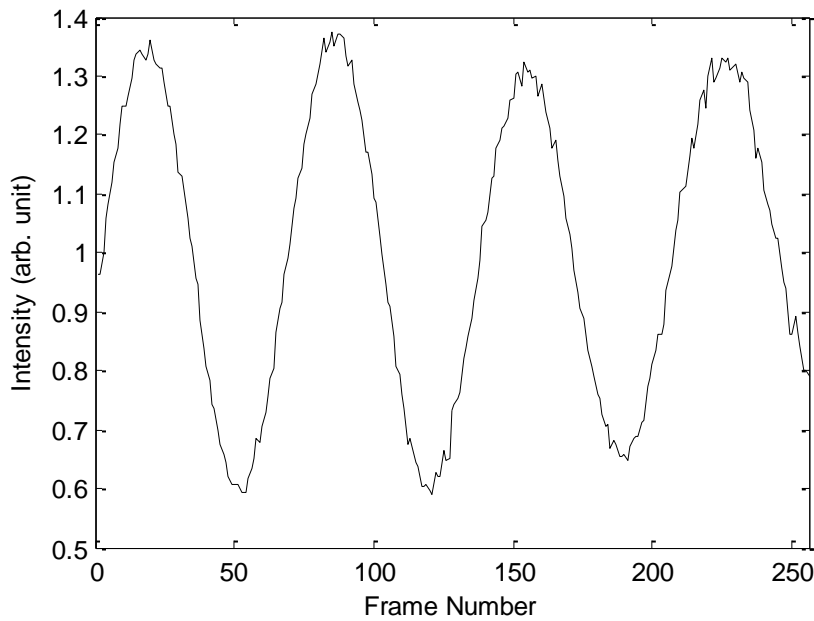


Figure 6.4 Compensated interference pattern

An FFT is applied to Equation 6.10 to find the spectrum of the pattern distribution. The spectrum of the FFT output contains three main terms as stated in Equation 6.11. The first term is a constant amplitude that relates to the light intensity in each interferometer arm, the second and third terms are related to phase shift achieved by scanning the wavelength of the white light. The purpose

of the FFT is to distinguish between the useful information which is represented by the phase change (i.e. c or c^* term) and the unwanted information of constant amplitude (i.e. A).

$$FFT[I_{xy}] = A(f) + C(f - f_o) + C^*(f + f_o) \quad (6.11)$$

f_o is a spatial frequency corresponding to the wavelength scanning and is a function of the optical path difference. The spectrum of Equation 6.11 is shown in Figure 6.5 and can be rewritten in a matrix form as shown in Equation 6.12.

$$FFT \begin{pmatrix} I(1) \\ I(2) \\ \vdots \\ \vdots \\ \vdots \\ I(n) \end{pmatrix}_{xy} = \begin{pmatrix} A(f) \\ \cdot \\ C(f - f_o) \\ \cdot \\ \cdot \\ C^*(f + f_o) \\ \cdot \end{pmatrix}_{xy} \quad (6.12)$$

The unwanted spectrum A and C^* are filtered out by replacing their values with zeros as shown in Equation 6.13 and Figure 6.6.

$$Filtration\ result = \begin{pmatrix} 0 \\ \cdot \\ C(f - f_o) \\ 0 \\ \cdot \\ 0 \\ 0 \end{pmatrix}_{xy} \quad (6.13)$$

The inverse FFT is applied to the filtered pattern to reconstruct the c value in Equation 6.10. Then a natural logarithm is applied to separate the phase from the exponential term and the unwanted amplitude variation as illustrated in Equations 6.14 and 6.15,

$$\ln(b_{xy} e^{j\phi(i)}) = \ln(b_{xy}) + j\phi(i) \quad (6.14)$$

$$\begin{pmatrix} I'(1) \\ I'(2) \\ \vdots \\ I'(n) \end{pmatrix}_{xy} = \ln(\text{ifft}) \begin{pmatrix} 0 \\ \cdot \\ C(f - f_o) \\ \cdot \\ 0 \\ 0 \\ 0 \end{pmatrix}_{xy} = \begin{pmatrix} \ln(0.5b(1)) \mp j\phi(1) \\ \ln(0.5b(2)) \mp j\phi(2) \\ \cdot \\ \cdot \\ \cdot \\ \ln(0.5b(n)) \mp j\phi(n) \end{pmatrix}_{xy} \quad (6.15)$$

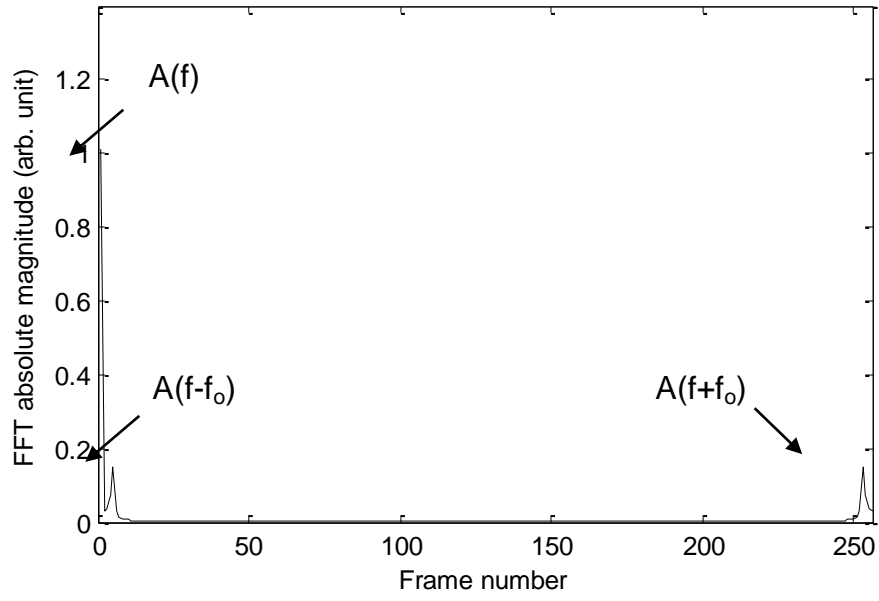


Figure 6.5 The Fourier transform spectrum of Equation 6.11

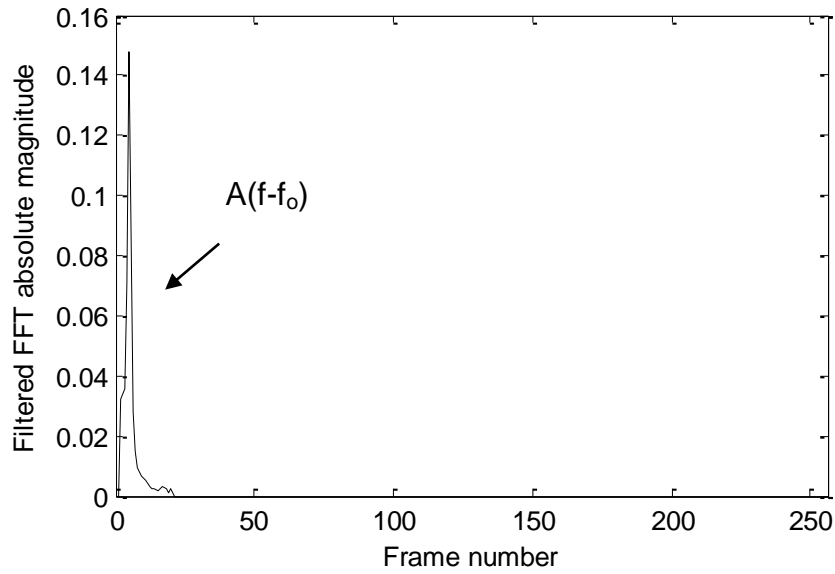


Figure 6.6 The Fourier transform spectrum of equation 6.13

The phase is extracted from the imaginary part of Equation 6.14, as shown in Figure 6.7. As the calculated phase is limited to a range of $-\pi$ to π , the phase distribution suffers from discontinuities. These discontinuities are corrected by adding a 2π staircase function to the phase distribution as phase unwrapping proceeds. This function is known as offset phase distribution and is shown in Figure 6.8.

The offset phase distribution is computed by monitoring the phase variation in Figure 6.7. Initially the offset phase was set to zero. The monitoring process was implemented by comparing each point in the discontinuities phase distribution to the one preceding it. Whenever there is 2π phase jump, 2π should be added to the offset point. This phase unwrapping process is described in the following condition statement:

$$\phi_{\text{offset}}(i) = \begin{cases} 0 & \text{if } i = 1 \\ \phi_{\text{offset}}(i-1) & \text{if } |\phi_{\text{offset}}(i) - \phi_{\text{offset}}(i-1)| < 2\pi \\ \phi_{\text{offset}}(i-1) + 2\pi & \text{if } |\phi_{\text{offset}}(i) - \phi_{\text{offset}}(i-1)| = 2\pi \end{cases} \quad (6.16)$$

After adding the offset phase to the discontinuous determined phase, the corrected final continuous phase is obtained as shown in Figure 6.9. The non-linear deviations existing at the edges of the phase distribution are compensated for by using linear least square fitting approximation, as shown in Figure 6.10.

Linear fitting using least squares is determined by making the variance S a minimum as follows.

$$S(a,b) = \sum_{i=1}^N (Y_i - (a + bx_i))^2 \quad (6.17)$$

where Y is the measured discrete data, N is the length of the discrete data (i.e. number of captured frames), $i=1,2,3,\dots,N$ is the frame number, x is the number of the discrete data. a and b are the coefficients of the linear equation to be fitted. The function $S(a,b)$ is minimum when $\partial S / \partial a = \partial S / \partial b = 0$. Thus,

$$\frac{\partial S}{\partial a} = \sum_{i=1}^N 2(Y_i - a - bx_i)(-1) = 0 \quad (6.18)$$

$$\frac{\partial S}{\partial b} = \sum_{i=1}^N 2(Y_i - a - bx_i)(-x_i) = 0 \quad (6.19)$$

Rearranging equations 6.18 and 6.19 yields,

$$aN + b \sum_{i=1}^N x_i = \sum_{i=1}^N Y_i \quad (6.20)$$

$$a \sum_{i=1}^N x_i + b \sum_{i=1}^N x_i^2 = \sum_{i=1}^N x_i Y_i \quad (6.21)$$

Evaluating the summations (i.e. $\sum x$, $\sum x^2$, $\sum Y$ and $\sum xY$) and substituting into equation 6.20 and 6.21 simplify the equations. Then the coefficient a and b can be solved by using the Gaussian elimination method.

Finally the height, which is equal to half of the optical path difference, is calculated from the slope of the fitted phase distribution using Equation 6.4.

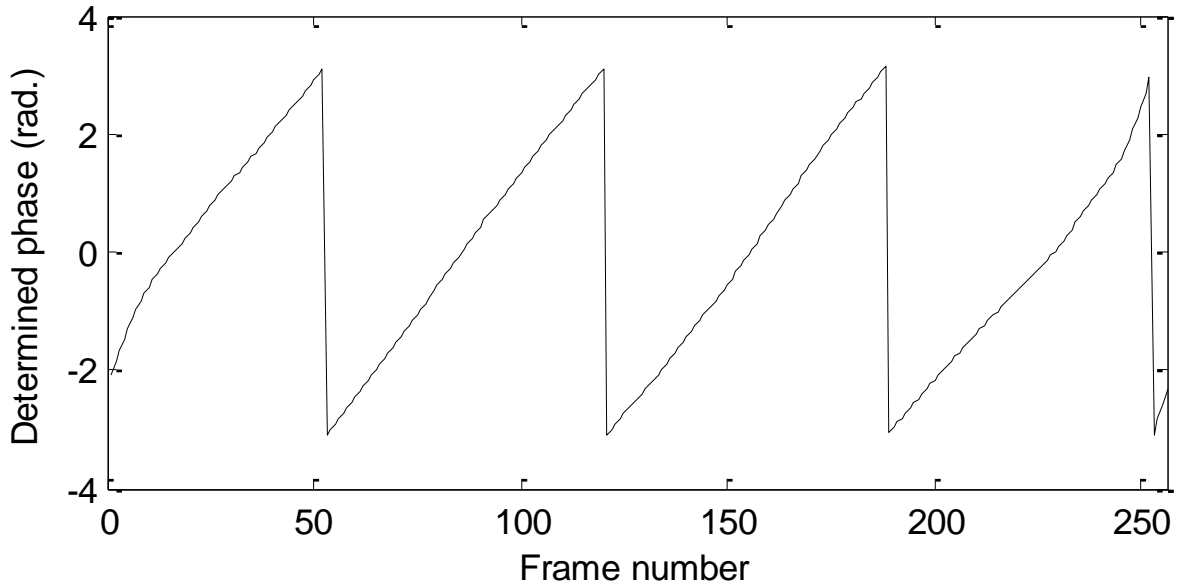


Figure 6.7 The discontinuous phase distribution as determined

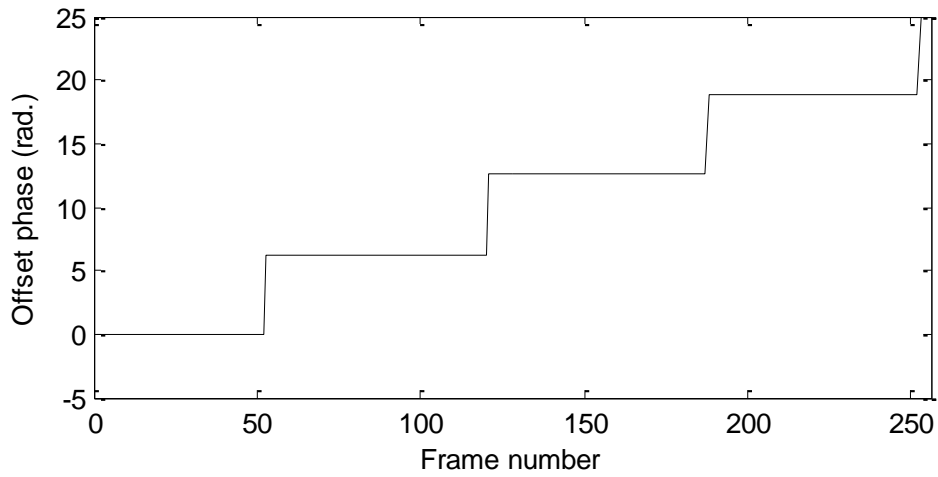


Figure 6.8 The offset phase distribution

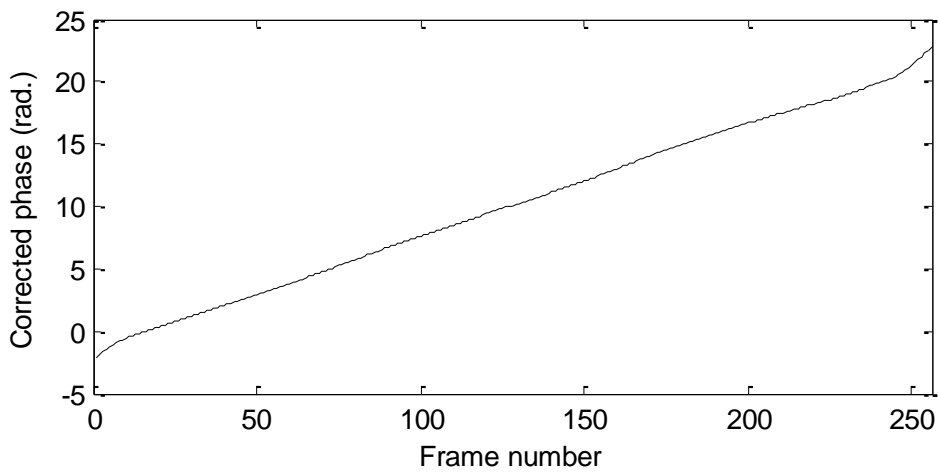


Figure 6.9 The corrected continuous phase distribution

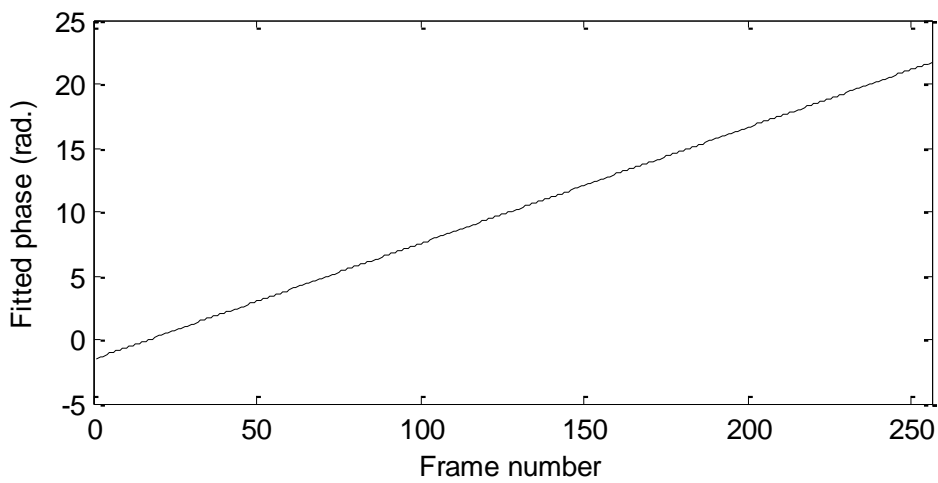


Figure 6.10 The fitted continuous phase distribution

The resolution at which the value of the phase shift may be determined, $\delta\phi$ in this algorithm, varies depending on the optical path difference. The following mathematical description can be used to evaluate the measurement resolution. The entire phase shift $\Delta\phi$ is assumed to be $2\pi (F+\epsilon)$ where the F is number of fringe cycles within the interference pattern and ϵ refers to the remaining fraction of the interference fringe pattern. As such, the phase resolution is,

$$\delta\phi = \frac{2\pi(F + \epsilon)}{N} \quad (6.22)$$

where N is the number of the captured frames, which equates to the total number of sampled points. The height resolution can be found by substituting Equation 6.22 into Equation 6.5 yielding,

$$\delta h = \left(\frac{F + \epsilon}{N}\right) \frac{\lambda_s}{2} \quad (6.23)$$

This equation shows that the resolution can be improved by factor $(F + \epsilon)/N$, compared to the simple FFT described in Section 6.2.2. For instance, if the surface height is located for example at $8.1 \mu\text{m}$ from the zero optical path difference then $F + \epsilon = 3.739$ and the resolution can be improved from $2.156 \mu\text{m}$ (using a simple FFT) to $0.031 \mu\text{m}$ for a given set of measurement parameters.

However, the resolution factor $(F + \epsilon)/N$ is dependent on the surface height so that, effectively, each pixel has a different resolution depending on its distance from the virtual reference.

6.3.4 Localised Peaks of Interference Pattern Using Convolution: Algorithm D

The principle of fringe analysis using a convolution is based on determining the positions of the peaks in the interference pattern according to the scanned wavelength. This method was introduced by Schwider (1994) using a dispersive interferometer. This kind of interferometer can be used to measure surface profile if a two-dimensional CCD camera is used. However, using the WSI

described in this thesis, areal measurement can be achieved. Furthermore, the resolution at which a specific surface height may be measured using the WSI can be improved by simply increasing the number of captured frames. The resolution of the dispersive interferometer is fixed because the number of sampling points is dependent on the number of CCD pixels available.

The surface height can be calculated by determining the number of peaks and their corresponding wavelength from the interference pattern. The position of the peaks can be found by convolving the interference pattern with a function having the following criteria,

$$f(x) = \begin{cases} -1 & 0 < x < B \\ 1 & B < x < 2B \end{cases} \quad (6.24)$$

where x is the number of points of per convolved function. The $2B$ value is the width of the convolved function $f(x)$ which, for optimum performance, should be approximately 74% of the width of the interference pattern as proposed by Snyder (1980).

The convolution process produces a filtered first derivative of the sinusoidal interference pattern with no DC bias. Thus neither further filtration nor background intensity compensation is required to further manipulate the resulting signal. Figure 6.11 illustrates the concept of the convolution process. It can be seen from this example that the area under the curve of the positive part is equal to the area at the negative part when the centre of $f(x)$ reaches the peak of the interference pattern. Hence, considering Figure 6.11, the net area under the curve after multiplication by $f(x)$ is zero when T_p is at the point on the abscissa corresponding to the peak on the curve.

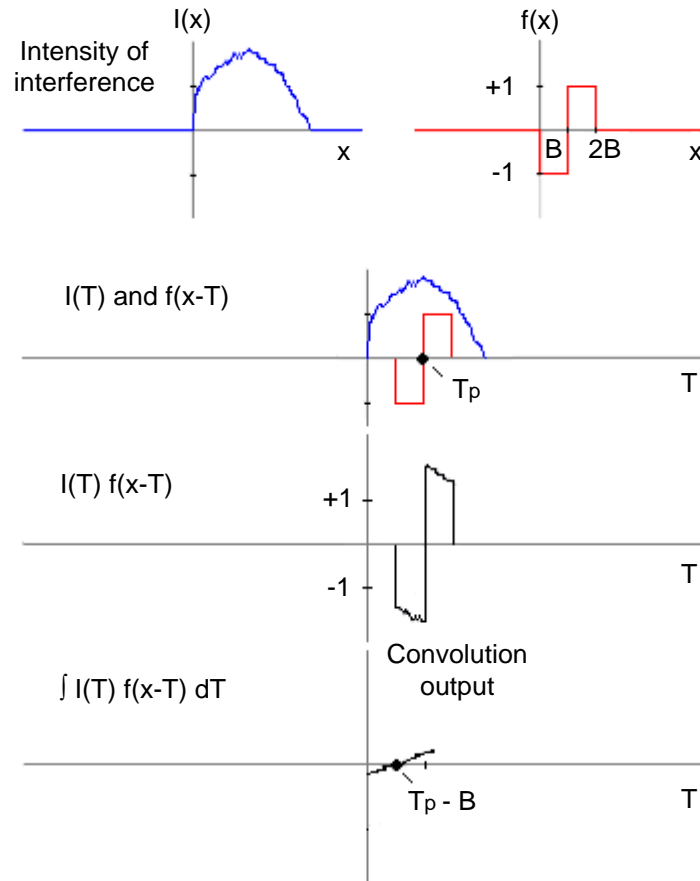


Figure 6.11 Convolution process

The obtained signal intersects the propagation axis at the original peaks positions as shown in Figure 6.12. Thus, by monitoring the convolved signal for a change of sign of the signal, the positions of the peaks can be determined (i.e. when the signal value is changed from negative to positive value a peak is present). However, these zero points are shifted from the original by a magnitude equal to the half of the width of $f(x)$ (i.e. shifted by the value of B) due to the effect of the convolution integration (see Appendix B.2). The direction of shifting is related to the sweeping start point of $f(x)$. If $f(x)$ starts sweeping from the first point at the interference signal, then the zero points are shifted to the right; but if the sweeping process starts at a point equal to the width of $f(x)$, then the zero points are shifted to the left which is the case in this analysis. This shifting process is known as zero position shifting.

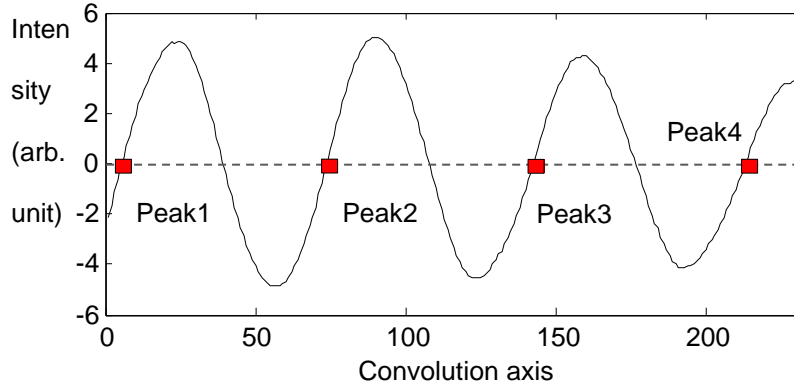


Figure 6.12 Convolution output

Since the phase shift difference between two successive peaks is 2π , the surface height can be calculated by,

$$h = \frac{(N_{pk} - 1)}{2 * (\frac{1}{\lambda_m} - \frac{1}{\lambda_n})} \quad (6.25)$$

where λ_m and λ_n are the wavelengths of the first and last peaks respectively and N_{pk} is the total number of peaks.

The following analysis can be used to evaluate the measurement resolution of this method. If the scanned wavelength range between λ_m and λ_n is $\Delta\lambda$, then Equation 6.25 can be written as,

$$h_1 = \frac{(N_{pk} - 1)}{2} \frac{\lambda_m^2 + \lambda_m \Delta\lambda}{\Delta\lambda} \quad (6.26)$$

The minimum height difference that can be resolved depends on the detection of the peak shift caused by a scanned wavelength step, $\delta\lambda$. The peak movement yields h_2 to be,

$$h_2 = \frac{(N_{pk} - 1)}{2} \frac{\lambda_m^2 + \lambda_m(\Delta\lambda - \delta\lambda)}{(\Delta\lambda - \delta\lambda)} \quad (6.27)$$

By subtracting Equations 6.26 from 6.27 the height resolution is found to be

$$\delta h = \frac{(N_{pk} - 1)}{2} \frac{\lambda_m^2 \delta\lambda}{\Delta\lambda(\Delta\lambda - \delta\lambda)} \quad (6.28)$$

Equation 6.28 shows that the resolution depends on the sampling frequency, which is contained in the parameters $\delta\lambda$ and N_{pk} . It is also dependent on the wavelength at which the first peak occurs, λ_m , as well as the wavelength range separation between the first and last peak, $\Delta\lambda$.

For example, the first and last peaks of the interference pattern in Figure 6.1(a) are found to be located at 600.15 nm and 675.02 nm respectively. Consequently $\lambda_m = 600.15$ nm and $\Delta\lambda = 74.87$ nm. Since this interference pattern contains 4 peaks, the resolution given by Equation 6.28 is 44.1 nm.

6.4 Experiment Results and Discussion

6.4.1 Measuring Sample Step Height

In order to confirm the performance of the algorithms discussed in the previous section the WSI measurement system, as illustrated in figure 5.1, was used to measure two step height standards having height values of 4.707 μm and 100 nm respectively. The measurement was carried out with 2X objective lenses.

Figures 6.13 to 6.16 show the four areal measurement evaluations of the 4.707 μm step height sample using the four previously described algorithms. The absolute distance of the surfaces, from the virtual reference mirror, and the sample step height values are listed in Table 6.1. The step height is determined by the difference between the mean line value of the upper and lower sample surfaces. The step height, as determined, is compared to the standard height value (4.707 μm) by finding the absolute difference between them which is noted as measurement error. The maximum height of the topographic surface parameter, S_{5z} is also given in the table. This parameter is found by calculating the average value of the absolute distance between five highest peaks and five lowest valleys within the surface. The computation time is also given in Table 6.1. The algorithms are

written in a Matlab environment and processed using a quad-core AMD Phenom™ II processor clocked at 2.5 GHz.

Table 6.1 Measured results for $h_r=4.707 \mu\text{m}$ step height standard using four algorithms

Measured parameters	Algorithm A (μm)	Algorithm B (μm)	Algorithm C (μm)	Algorithm D (μm)
Mean of LS height	8.663	8.358	8.150	8.161
Mean of US height	12.995	12.914	12.846	12.852
S_{5z} in LS	0	0.012	0.021	0.030
S_{5z} in US	0	0.028	0.030	0.042
Measured step height (h_s)	4.332	4.556	4.696	4.691
Measurement error = $ h_r - h_s $	0.375	0.151	0.011	0.016
Computation time (sec)	11.63	100.14	31.99	19.43

The LS and US are the lower and upper surfaces respectively

The results show that the measurement error in the step height measurement decreased from $0.375 \mu\text{m}$ to $0.011 \mu\text{m}$. This variation in the measurement accuracy obviously returns to the algorithm computing resolution. Except for algorithm A, the areal measurement results contain surface roughness information, due to the enhancement in measurement resolution. This enhancement also results in detection of low level noise such as environmental disturbances, which may cause additional measurement error. Table 6.1 shows that the S_{5z} values for the lower surface are less than the upper surface. This could be due to the variation in the measurement resolution for different heights as illustrated in Equations 6.14 and 6.19.

The computing speed is influenced by the complexity of the algorithms. Clearly the simple FFT is the fastest but has low accuracy. However, the fringe analysis using FFT has moderate speed but the highest accuracy.

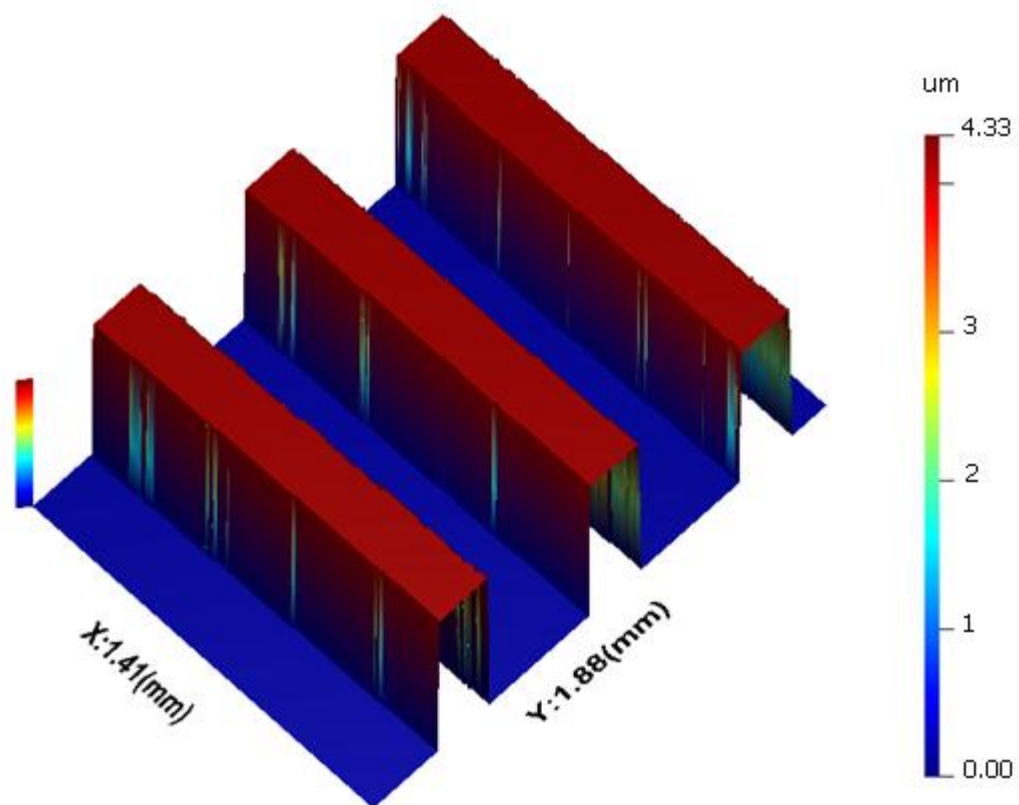
The same analysis algorithms were used to evaluate the 100 nm step height sample as shown in Table 6.2. The areal measurements and cross section profiles are shown in figures 6.17, 6.18 and 6.19. The step height is determined by the difference between the mean line value of the upper and lower sample profiles. Algorithm A was unable to recognize the step height while algorithm B can distinguish the step height of 100 nm but with error 84 nm as shown in figure 6.17(b). This error could be due to the limited resolution of algorithm B. The algorithms C and D found the step height to be 100.8 nm and 102.3 nm respectively as shown in Figures 6.18(b) and figure 6.19(b) respectively. The S_{5z} values on the two surfaces are the same for each algorithm. This is because the difference between the interference patterns of both surfaces is minimal; the fringe period sampling points of both patterns are almost the same. Therefore, maintaining the sampling of the interference patterns can sustain the resolution.

Table 6.2 Measured results for $h_r=100$ nm step height standard using four algorithms

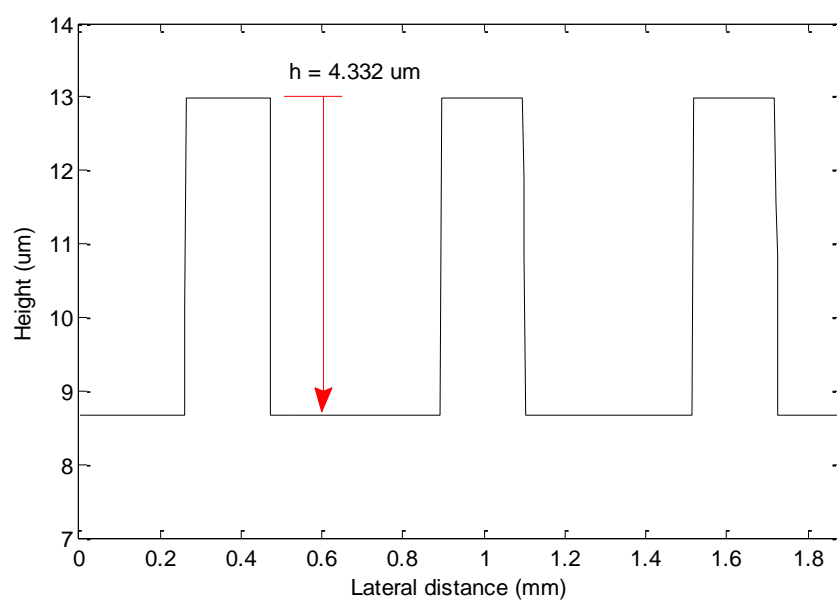
Measured parameters	Algorithm A (μm)	Algorithm B (μm)	Algorithm C (μm)	Algorithm D (μm)
Mean of LS height	16.31	17.11	16.675	16.916
Mean of US height	16.31	16.926	16.574	16.8137
S_{5z} in LS	0	0.013	0.012	0.014
S_{5z} in US	0	0.013	0.013	0.014
Measured step height (h_s)	N/A	183	100.8	102.3
Measurement error = $ h_r - h_s $	N/A	0.083	0.0008	0.0023

The LS and US are the lower and upper surfaces respectively

Another two samples, a 9.758 μm step height and 0.78 μm sine wave were measured using WSI. The results are reported in Appendix B (3 & 4).

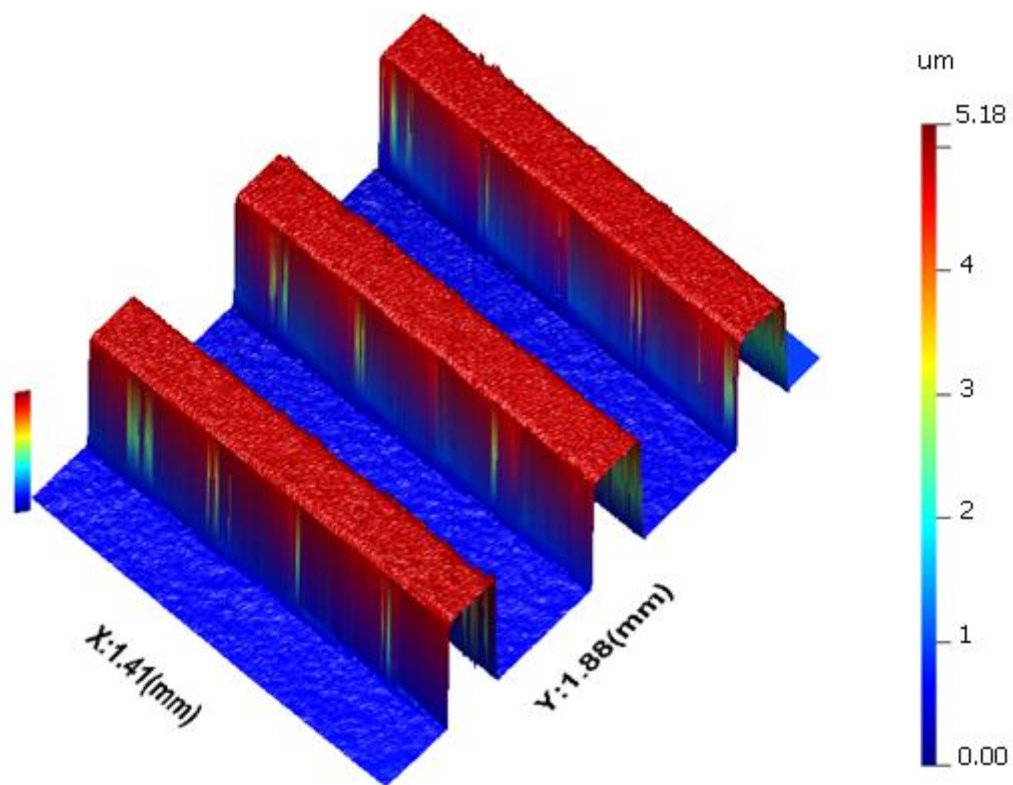


(a)

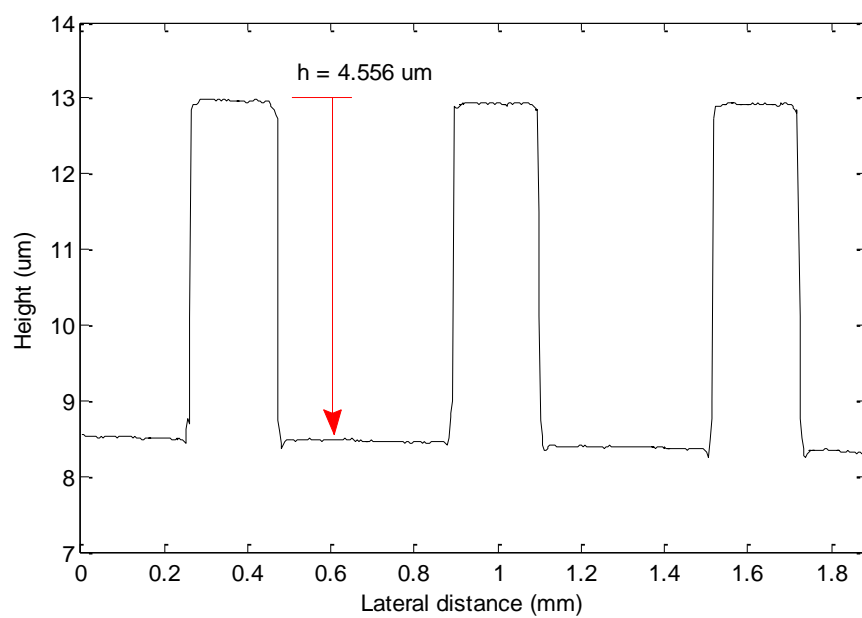


(b)

Figure 6.13 Measured results for 4.707 μm standard step height sample using algorithm A (a) areal measurement (b) cross-sectional profile

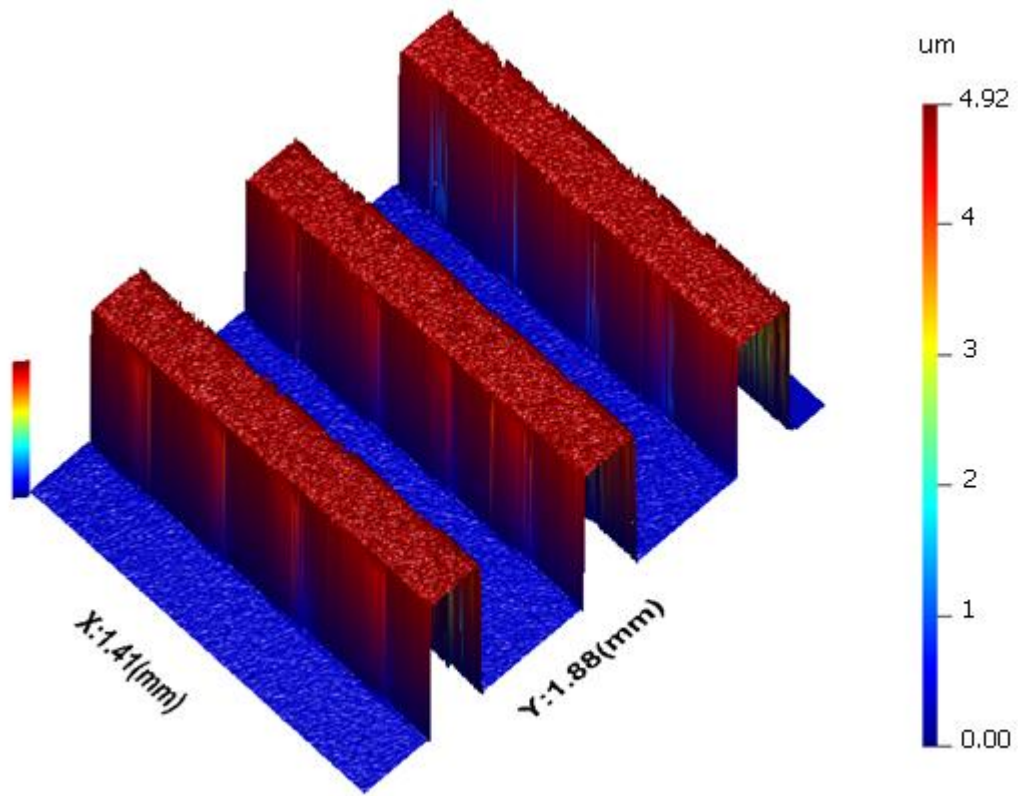


(a)

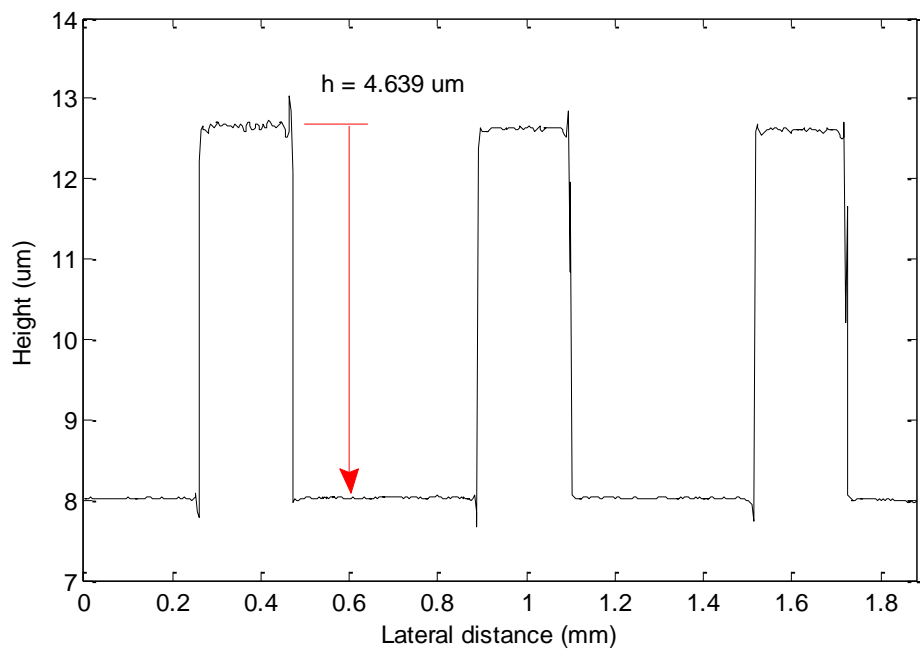


(b)

Figure 6.14 Measured results for 4.707 μm standard step height sample using algorithm B (a) areal measurement (b) cross-sectional profile

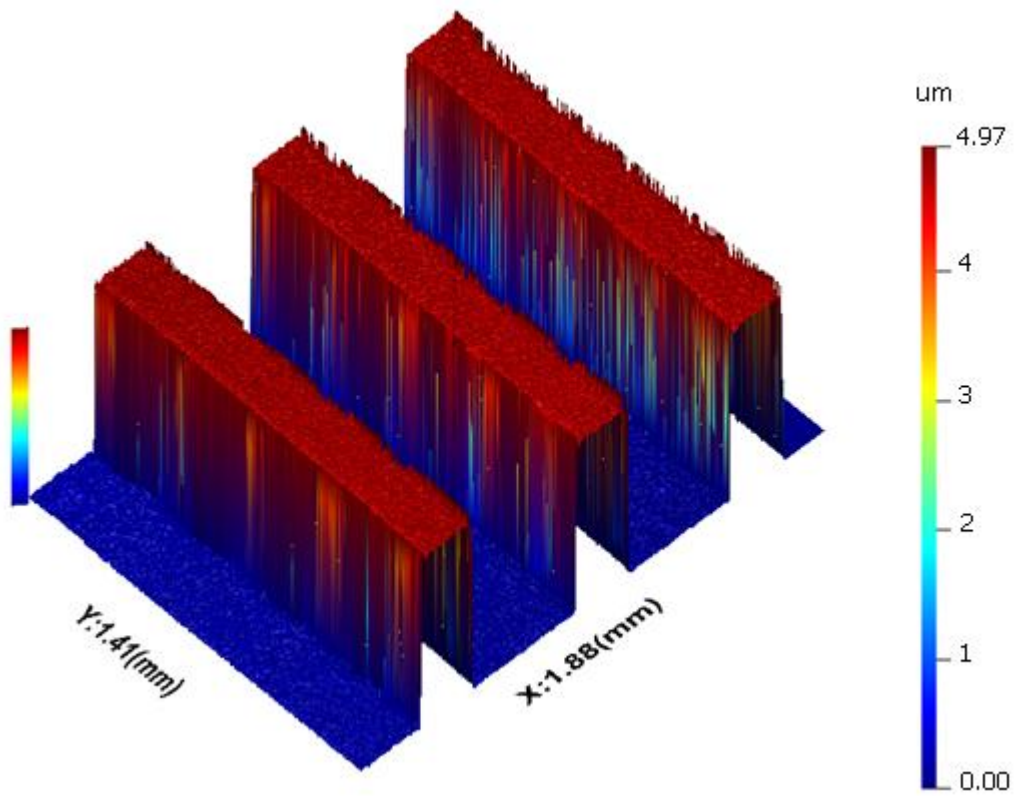


(a)

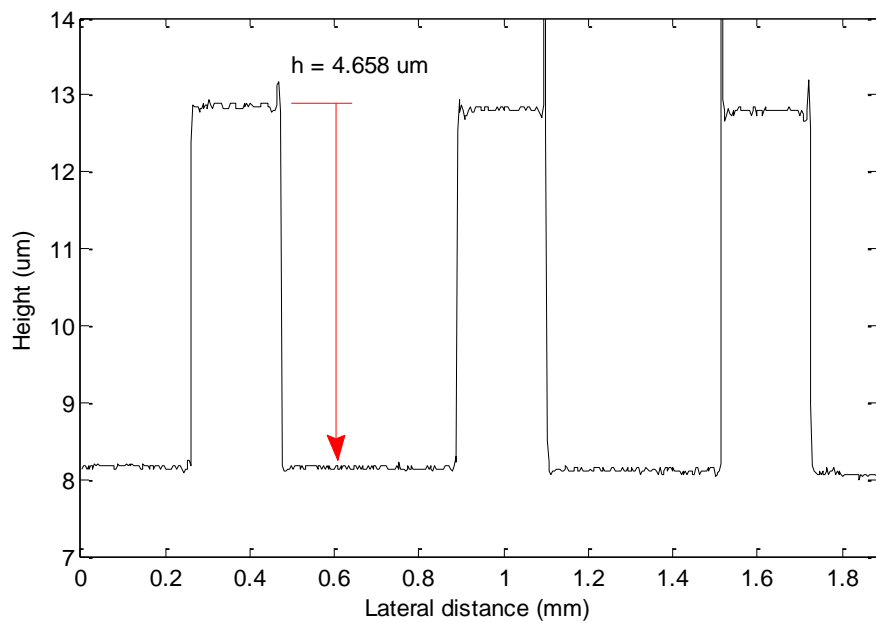


(b)

Figure 6.15 Measured results for $4.707 \mu\text{m}$ standard step height sample using algorithm C (a) areal measurement (b) cross-sectional profile



(a)



(b)

Figure 6.16 Measured results for 4.707 μm standard step height sample using algorithm D (a) areal measurement (b) cross-sectional profile

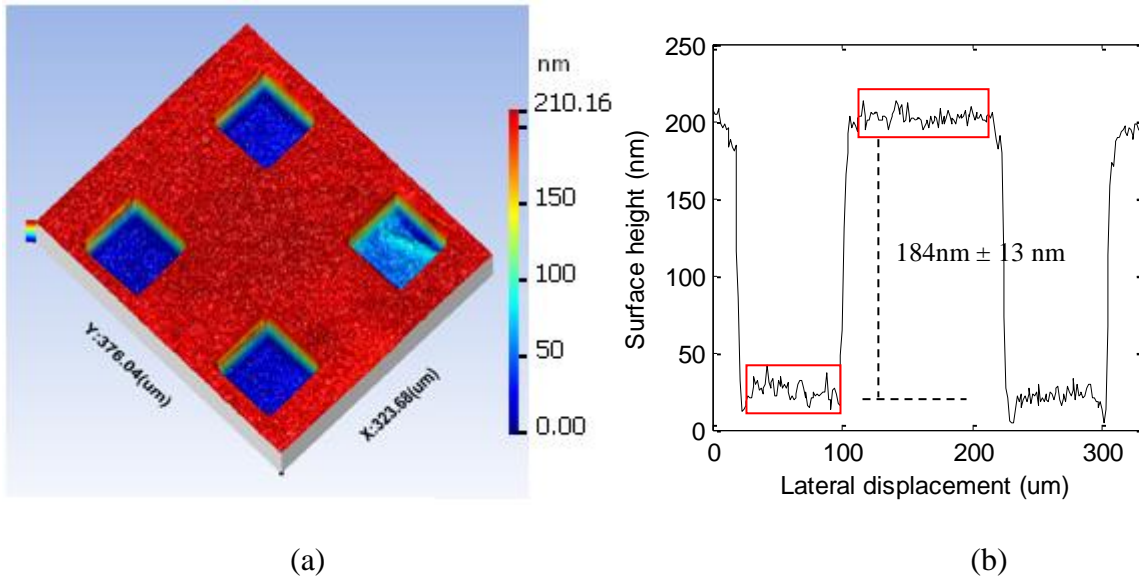


Figure 6.17 Measured result for 100nm standard step sample using algorithm B (a) areal measurement (b) cross-sectional profile

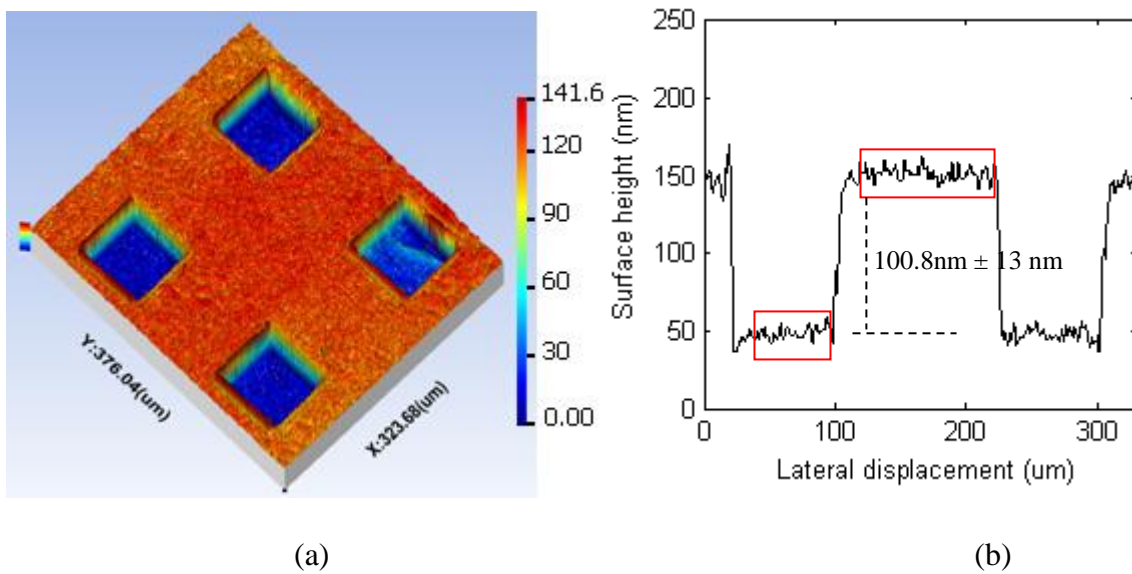


Figure 6.18 Measured result for 100nm step height standard sample using algorithm C (a) areal measurement (b) cross-sectional profile

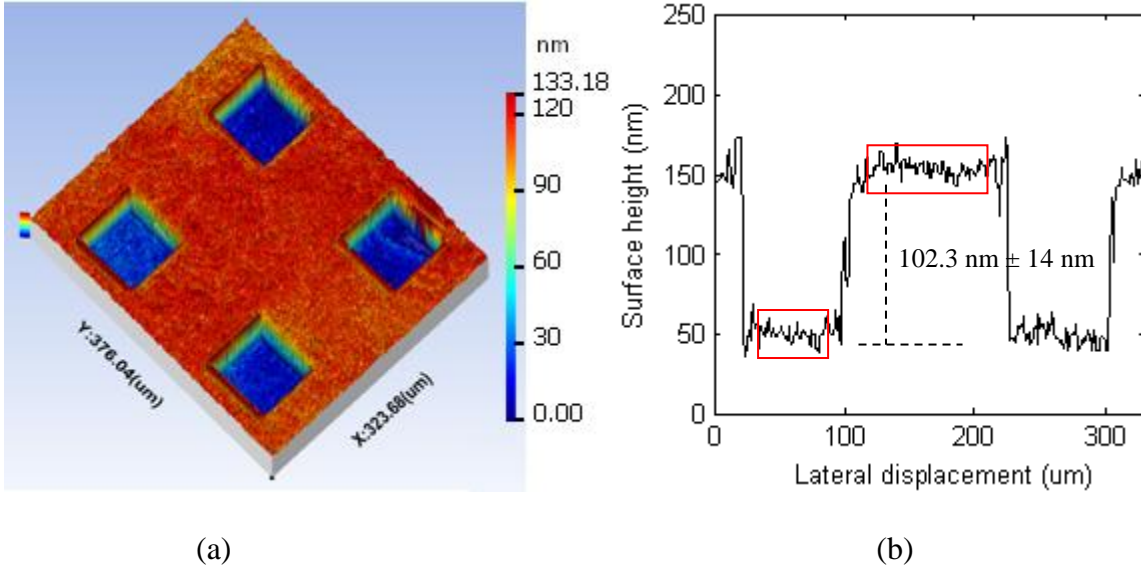


Figure 6.19 Measured result for 100nm step height standard sample using algorithm D (a) areal measurement (b) cross-sectional profile

6.4.2 Measuring Sample V Groove

One of main measurement limitations in interferometers is the surface gradient (Kaplonek and Lukianowicz, 2012, Gao et al., 2008). If the surface gradient exceeds the half of the angular angle of the objective lens, no interference fringes can be detected. For example, the maximum surface gradient is approximately 8° for a 5X objective lens with 0.14 NA as shown in the following calculation.

$$\theta_{\max} = \beta = \arcsin \frac{NA}{n} \quad (6.29)$$

$$\theta_{\max} = \arcsin \frac{0.14}{1} \cong 8^\circ$$

where θ_{\max} is the maximum acceptable surface gradient, β half the maximum value of the angular angle of the objective lens, NA is the numerical aperture and n is the refractive index of air. For large dimension samples with surface gradient exceeding the acceptance inclination angle, the

interferometers need to be rotated and tilted to decrease the surface gradient and satisfy the acceptance inclination condition.

In contrast to VSWLI and PSI interferometers, which are not suitable for such a solution, the WSI can be rotated and tilted to reduce the surface gradient because the measurement principle contain neither mechanical movement nor phase ambiguity issues.

The V groove sample, shown in Figure 6.20, is measured with the WSI using a 5X objective lens. This sample has a step height of $23.2\text{ }\mu\text{m}$ and an angle equal to 45° . The WSI is rotated by 45° from the vertical axis using a rotary stage to face the groove side walls as shown in Figure 6.21. The interferometer and the sample are correctly aligned when a single fringe is spread over the groove side walls. Figure 6.22 shows the interference fringe obtained from one side of the groove five walls. The remaining grooves show no interference fringes because of the lack of coherence length. The depth of focus has a limited range of $14\text{ }\mu\text{m}$ and several steps were detected, however some of the grooves were out of focus.

The measured result, shown in Figure 6.23, allows five steps to be evaluated. Three out of five (steps 3, 4 and 5) are in the opposite direction according to the reference mirror plane. Thus the steps suffer from height ambiguity (direction ambiguity) because the measured heights present absolute of the cross-sectional profile as shown in Figure 6.23(b). The height ambiguity was resolved by unwrapping the surface using prior knowledge of the sample structure. A surface mapping profile was constructed by monitoring the surface steps values. The middle step gives a value less than expected when compared to step 2, but gives a correct value when compared to step 4. Therefore, the surface mapping profile is set to unit positive value for steps 1 and 2, but to negative unit for steps 3, 4 and 5, see Figure 6.24. By multiplying the measured profile by the mapping profile, the corrected cross profile is obtained as shown in Figure 6.25.

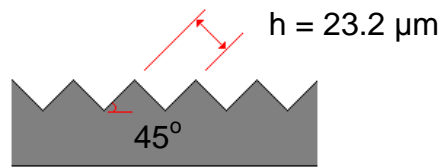


Figure 6.20 V groove sample



Figure 6.21 The WSI interferometer rotated by 45° to measure V groove sample

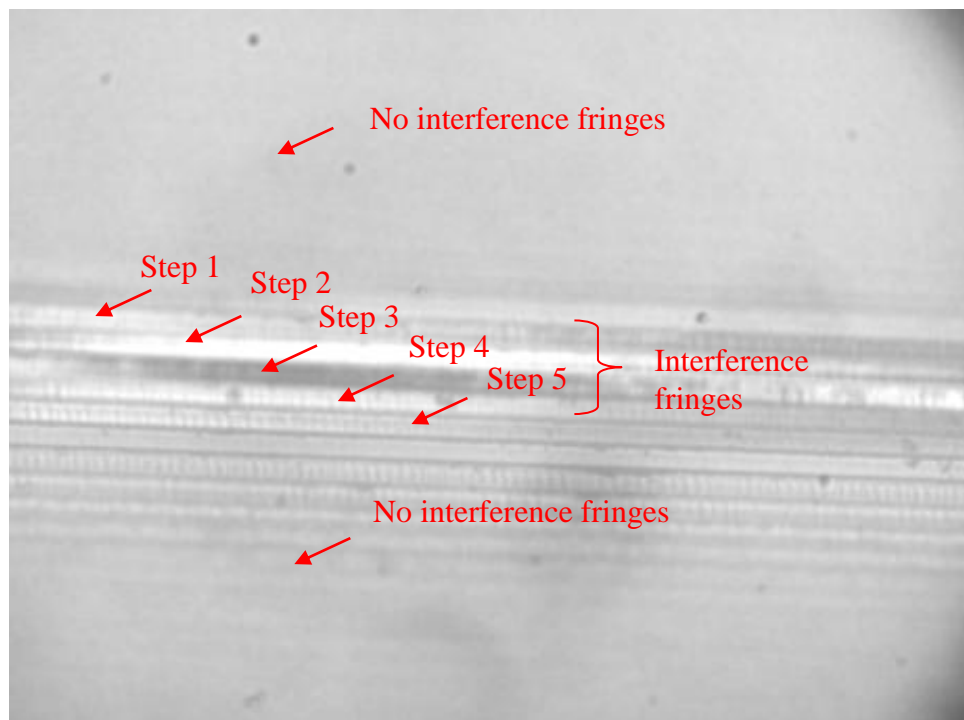
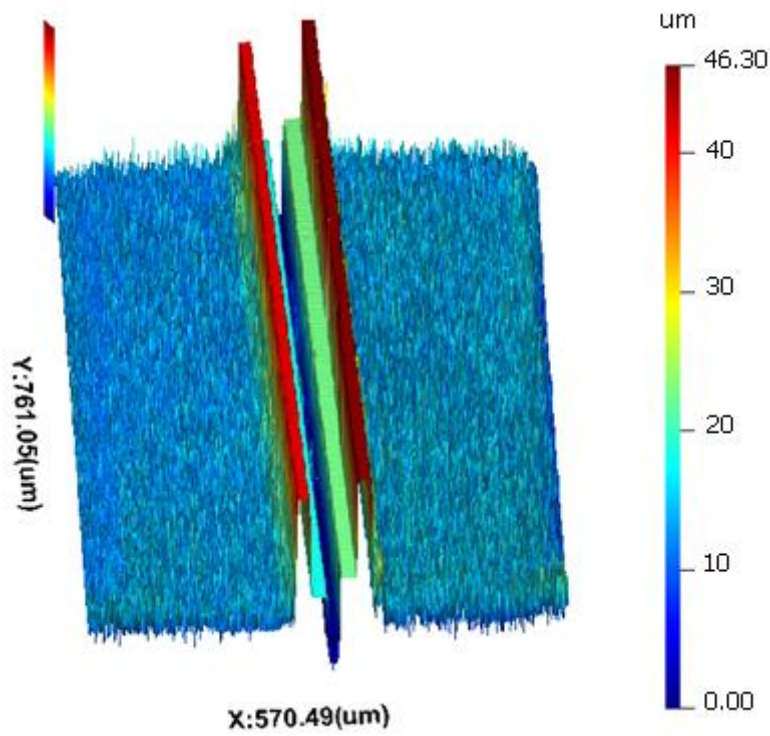
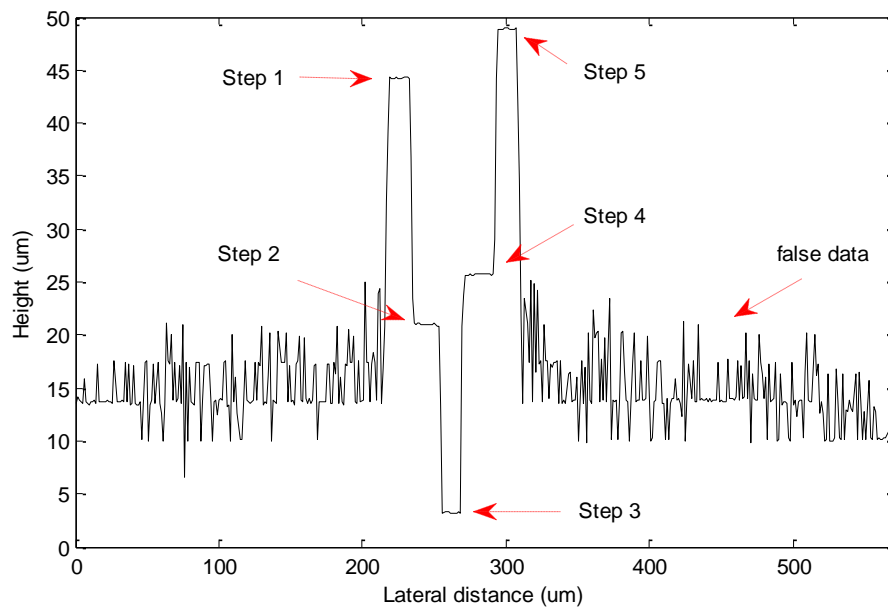


Figure 6.22 Captured frame showing 5 steps return of V groove sample



(a)



(b)

Figure 6.23 Measured results for V groove sample (a) areal measurement (b) cross-sectional profile suffers direction ambiguity

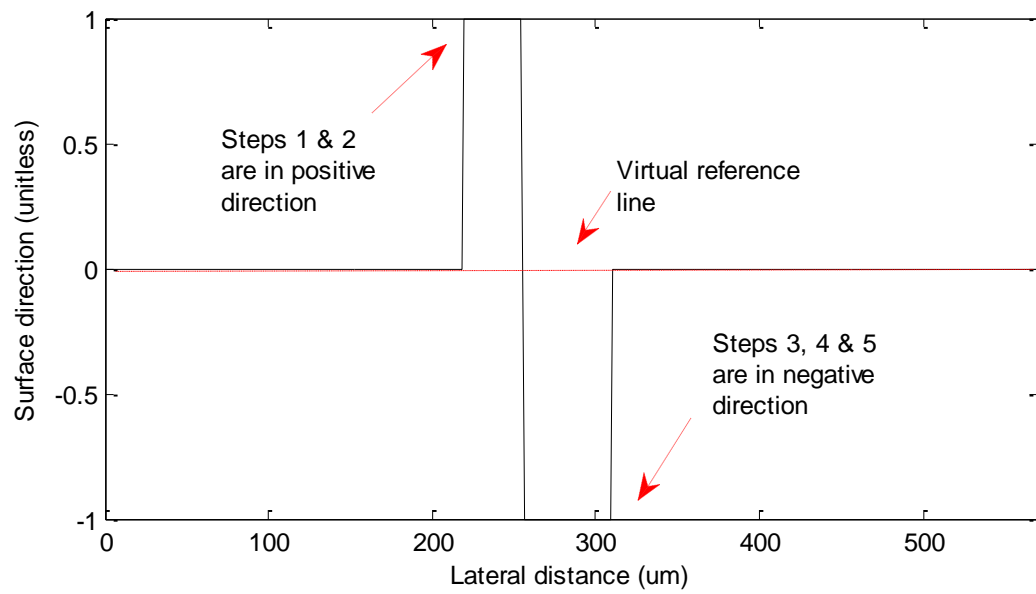


Figure 6.24 Mapping for step height direction

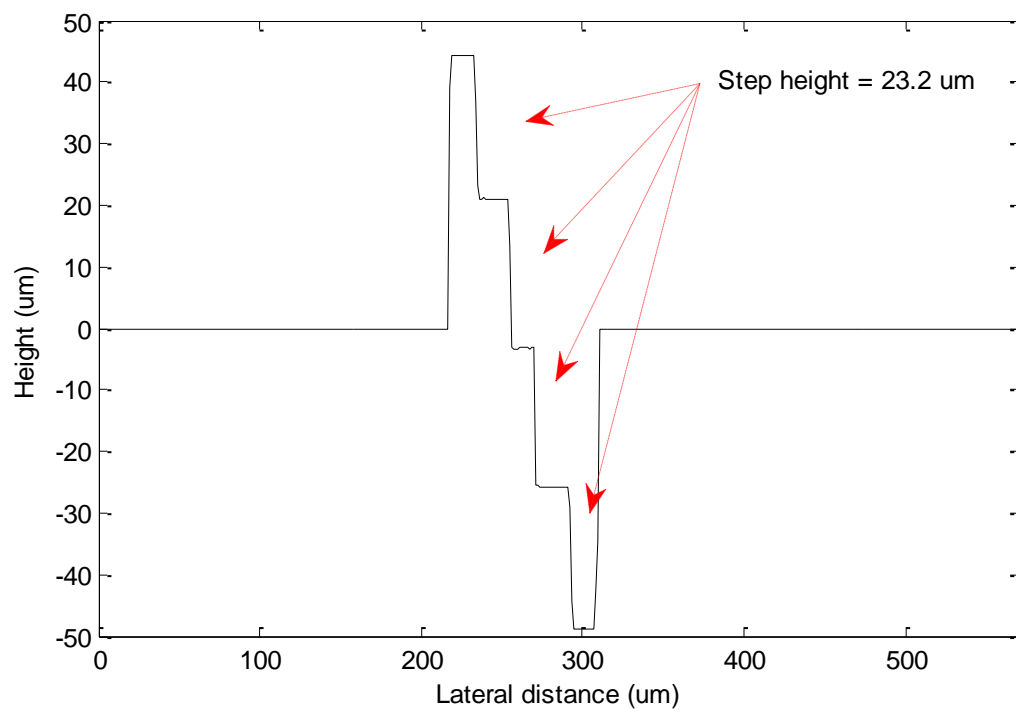


Figure 6.25 Corrected cross-sectional profile

6.5 Measurement Repeatability

The 4.707 μm step height standard sample was placed approximately at 15 μm out away from the virtual reference plane and measured 14 times to determine repeatability using 5X objective lenses as stated in Table 6.3. The time duration between each measurement was approximately 30 seconds. In order to find the measurement consistency, the standard deviation was calculated,

$$s_N = \sqrt{\frac{1}{N-1} \sum_{i=1}^N (x_i - \bar{x})^2} \quad (6.30)$$

where x_i is the measured value of the i^{th} sample, N is the total number of the measurement, and \bar{x} is the mean value of these measurements where it is found to be equal to 4.7189 μm . The calculated standard deviation value is equal to $s_N = 6.18 \text{ nm}$. The upper ($L_p = 4.7313 \mu\text{m}$) and lower ($L_l = 4.7065 \mu\text{m}$) limits are assumed to have separation of two standard deviations from the mean value and calculated by the equations,

$$L_p = \bar{x} + 2s_N \quad (6.31)$$

$$L_l = \bar{x} - 2s_N \quad (6.32)$$

Figure 6.26 shows that 9 measurements out of 14 are within one standard deviation separation while all the data, apart from point 11th, are within the upper and lower limits. The maximum deviation from the mean value (the eleventh point) was 13.9 nm which is 1.54 nm away from the lower limit. This deviation could be caused by environmental disturbances in the laboratory, since the laboratory containing the experiments is neither a cleanroom nor a temperature and vibration controlled room. This evaluation shows that the repeatability of the WSI is within two standard deviation separations, $2s_N = 12.36 \text{ nm}$.

Table 6.3 Measurement repeatability for the WSI using 4.707 μm step height sample

Number of measurement	Measured Step height value (μm)	Number of measurement	Measured step height value (μm)
1	4.72	8	4.718
2	4.723	9	4.718
3	4.714	10	4.727
4	4.717	11	4.705
5	4.719	12	4.71
6	4.726	13	4.726
7	4.72	14	4.722

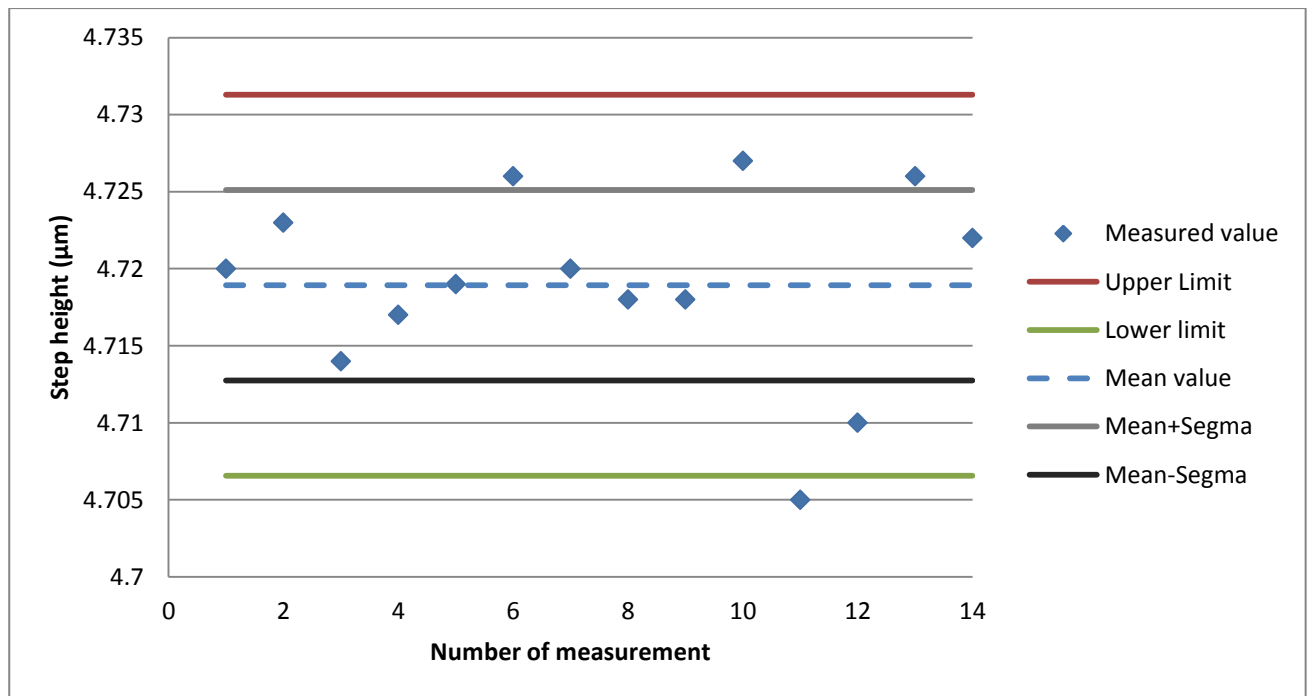


Figure 6.26 Measurement repeatability

6.6 Summary

This chapter introduces a high resolution WSI interferometer with four algorithms to analyse the measured data. The resolution can be improved from the micrometre to nanometre scale by changing the algorithm used for analysis. The resolution is increased from 2.165 μm using simple FFT to less than 0.1 μm using FFT fringe analysis. These results show that the measurement resolution depends on the optical path difference, wavelength scanning range and step height.

It was found that the resolution increased with surfaces that were close to the zero optical path difference. The S_{5z} values of surface texture were reduced by increasing the measurement resolution. However, if the analysis algorithm has a low computing resolution such as with the simple FFT no surface roughness information is observed.

The computing speed is influenced by the complexity of the algorithms. The FFT fringe analysis gives the best accuracy with moderate computing speed.

It is found that the fringe analysis method is more convenient to use and has a high resolution computing algorithm. Peaks identification using the convolution algorithm can give comparable measurement resolution when the width of the convolve function $f(x)$ is 74% that of the interference pattern.

In the following chapter, only the algorithm using the FFT fringe analysis will be considered for use with parallel programming. The flow chart in Figure 6.27 summarizes the FFT fringe analysis algorithm described in Section 6.2.4, when sequential programming is employed to evaluate areal measurement.

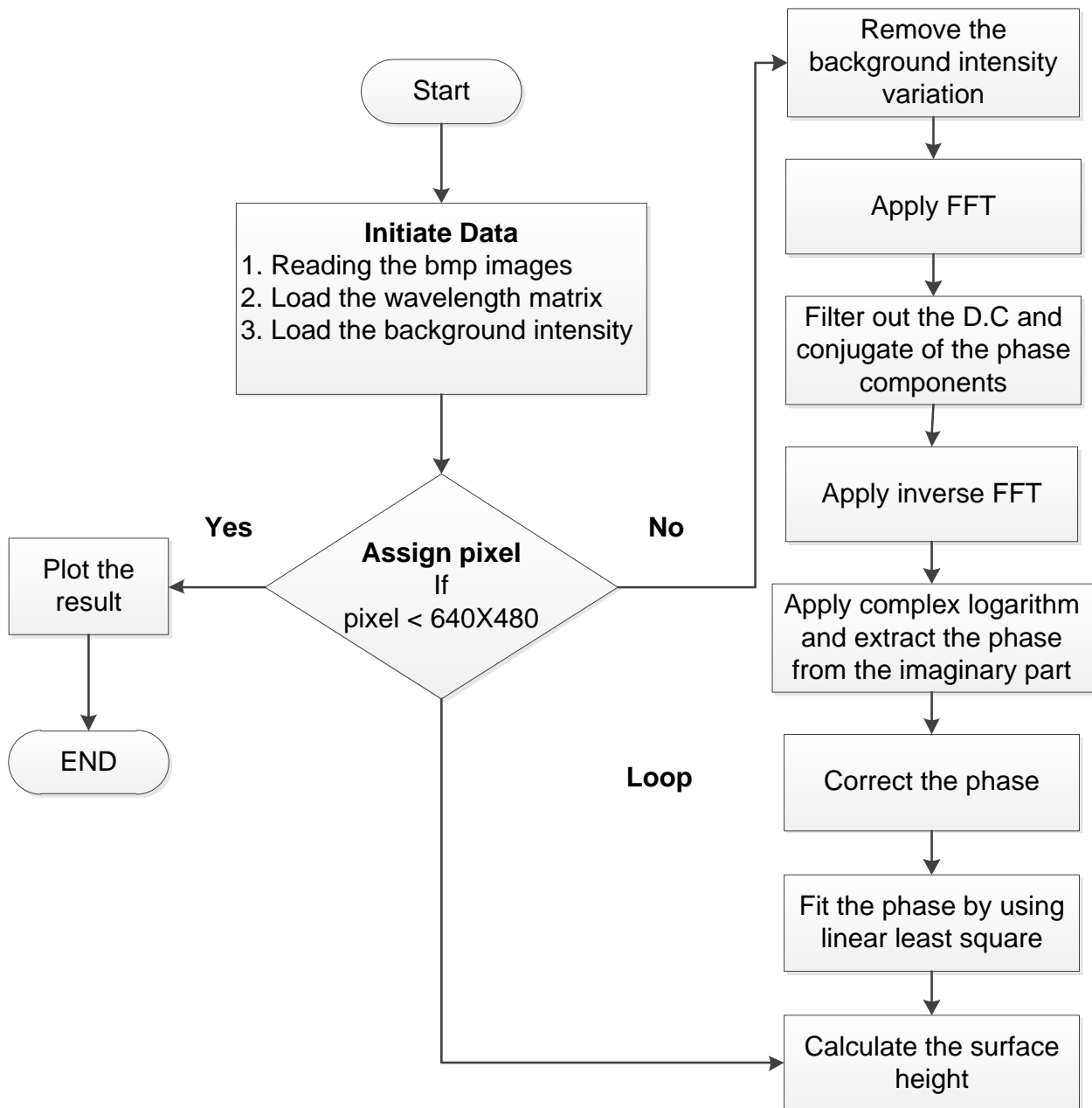


Figure 6.27 Flow chart of fringe analysis using FFT (algorithm C)

7. Accelerating Computing Process Using GPU

7.1 Introduction

In order to obtain the areal topography for the measured sample, the algorithms described in Chapter 6 should be applied across all the captured points. To find the areal topography of a sample viewed by 640 x 480 CCD pixels, the data analysis needs to be executed 307200 times, which is equal to the number of CCD pixels, in a sequential manner if a tradition C or Matlab code is used with multicore CPUs. It was found in Chapter 6 that the time required to analysis 256 frames is approximately 32 seconds if the FFT fringe analysis algorithm was applied. Using many GPU cores to evaluate the data in a parallel manner can significantly reduce the measurement throughput. This chapter presents a parallel programming model which can accelerate the computing process using Nvidia GTX285 GPUs.

7.2 GPUs versus CPUs

The general development of GPUs in last decade has been in part to accelerate graphic functions (i.e. fixed-function pipelines) which are vital for the video game industry (Owens et al., 2008). However, these functions were configurable but not programmable. Nevertheless, market demand for speedup of execution time and high-quality real time graphics has driven the GPUs designers to develop hardware, make it programmable and support floating point operation, as can be seen in later GPU specifications (NVIDIA, 2012a). Figure 7.1 illustrates the floating point operation per second (GFLOP/s) of NVIDIA GPUs versus Intel CPUs. The GTX 285 (which has the same number of processing cores and almost the same memory specification compared to GTX 280) can increase the operation rate of single precision floating point up to 100 times compared to the Intel Harpertown CPU (Kirk and Hwu, 2010, NVIDIA, 2011, NVIDIA², 2012, NVIDIA, 2012b).

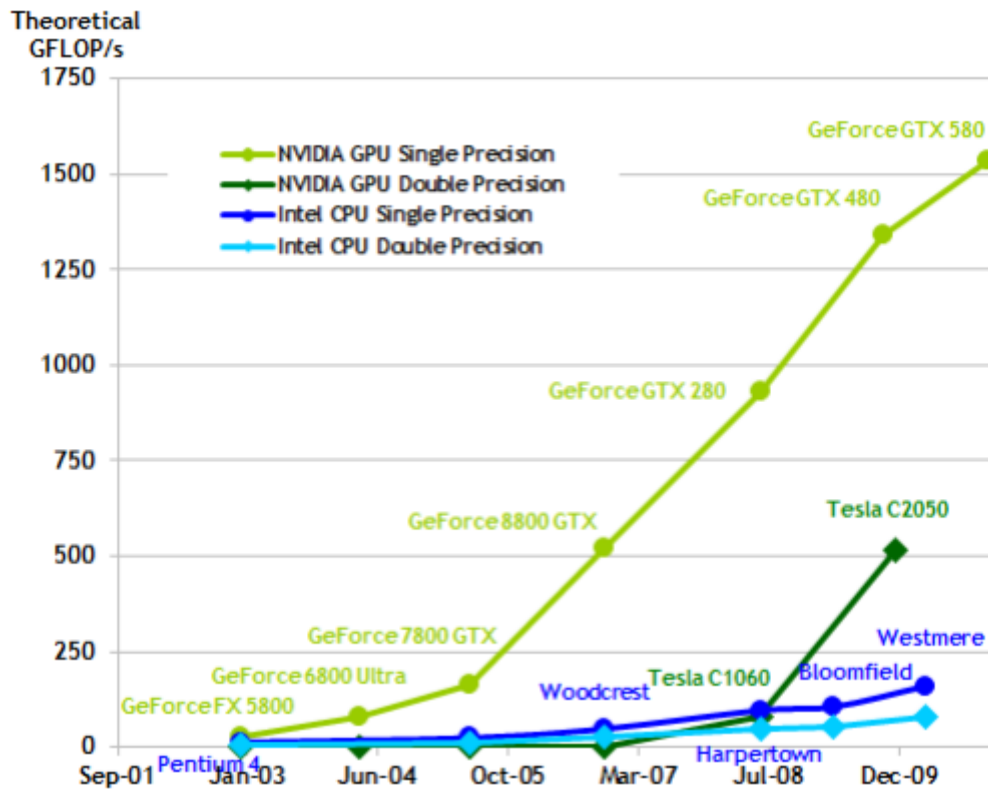


Figure 7.1 Floating point operation per second[adapted from (NVIDIA, 2011)]

The main reason for this acceleration is the many core trajectory in the GPUs being used for parallel execution applications. The design of GPUs is optimised for parallel code performance, using a massive number of threads and having a large memory bandwidth (159 GB/s for GTX 285). These aspects make GPUs an ideal solution for data parallelism.

Data parallelism performance is increased by increasing the number of cores (i.e. processors). As such, a parallel programming model is needed to scale the data parallelism into the number of cores. Furthermore this programming model needs to support both CPU/GPU execution of an application since most of the applications are based on both sequential and parallel data processing. For instance, the data capture during wavelength scanning in the WSI is achieved in a sequential manner while the data processing needs to be analysed in a parallel manner.

These requirements are achieved by a parallel programming model invented by NVIDIA at the end of 2006 known as CUDATM (Compute Unified Device Architecture).

7.3 Compute Unified Device Architecture

In 2006, NVIDIA introduced CUDA for general purpose computation using GPUs (McClanahan, 2010). With CUDA, scientists and researchers began to use the powerful floating point performance available using GPUs to process data for applications such as analysing air traffic flow and visualising molecules.

CUDA is a programming model that gives a link between the hardware architecture of the GPU and a high level programming language such as C/C++. This allow the programmer to access the GPU memory spaces and do arithmetic operation with single or double floating point precision (Whitepaper, 2009).

The CUDA program structure consists of host (CPU) code and device (GPU) code (see Figure 7.2) (Kirk and Hwu, 2010). The host code is standard ANSI C code while the device code is ANSI C code extended with data parallelising constructs. The extension is introduced by keywords and symbols to classify the device code from the host code (NVIDIA, 2011). For instance, CUDA extends the C function call syntax with <<< and >>> brackets to define execution configuration parameters. These parameters refer to the number and organisation of threads as seen in Appendix C.4 step 3. The extended C function in CUDA is known as a kernel. The kernel declaration syntax also extended by `__global__` keyword to instruct the compiler to call the kernel from the host and execute on the GPU as demonstrated in the kernel declaration in Appendix C.5.

The host code (i.e. C functions) is executed on the CPU in a conventional manner by a single thread using multi-core CPUs (e.g. dual cores). In contrast, the device code is executed on the GPU (including kernel functions) in a parallel manner by a massive number of generated threads using many core processors.

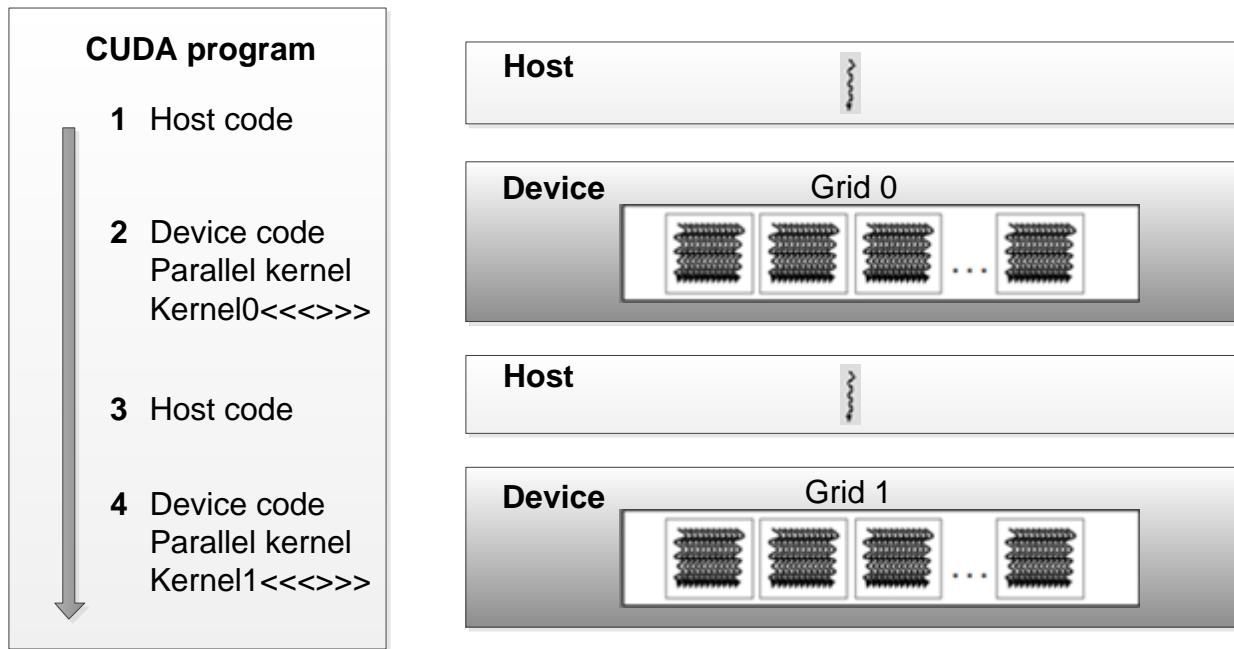


Figure 7.2 General CUDA program structure

Before introducing the structure of the proposed CUDA program two topics, thread hierarchy and memory hierarchy, are first introduced in Sections 7.3.1 and 7.3.2 respectively to give a better understanding of the CUDA structure.

7.3.1 Thread hierarchy

The acceleration in the computing process using CUDA is achieved by generating a massive number of threads that execute the device code on GPU unified processors (cores). The generated threads are organised in a grid with a two-level hierarchy (see Figure 7.3). The first level is a grid consisting of a two dimensional layout of thread blocks. Each block has a unique two dimensional coordinate ID given by CUDA built-in variables: `blockIdx.x` and `blockIdx.y`. In the second level, each thread block consists of a three-dimensional array of threads with a maximum size of 512 threads. Each thread in the block has a unique CUDA built-in three dimensional index: `threadIdx.x`, `threadIdx.y`, and `threadIdx.z`.

This thread hierarchy gives the programmer the ability to direct each thread to a specific part in the measured data containing the intensity values of a certain pixel.

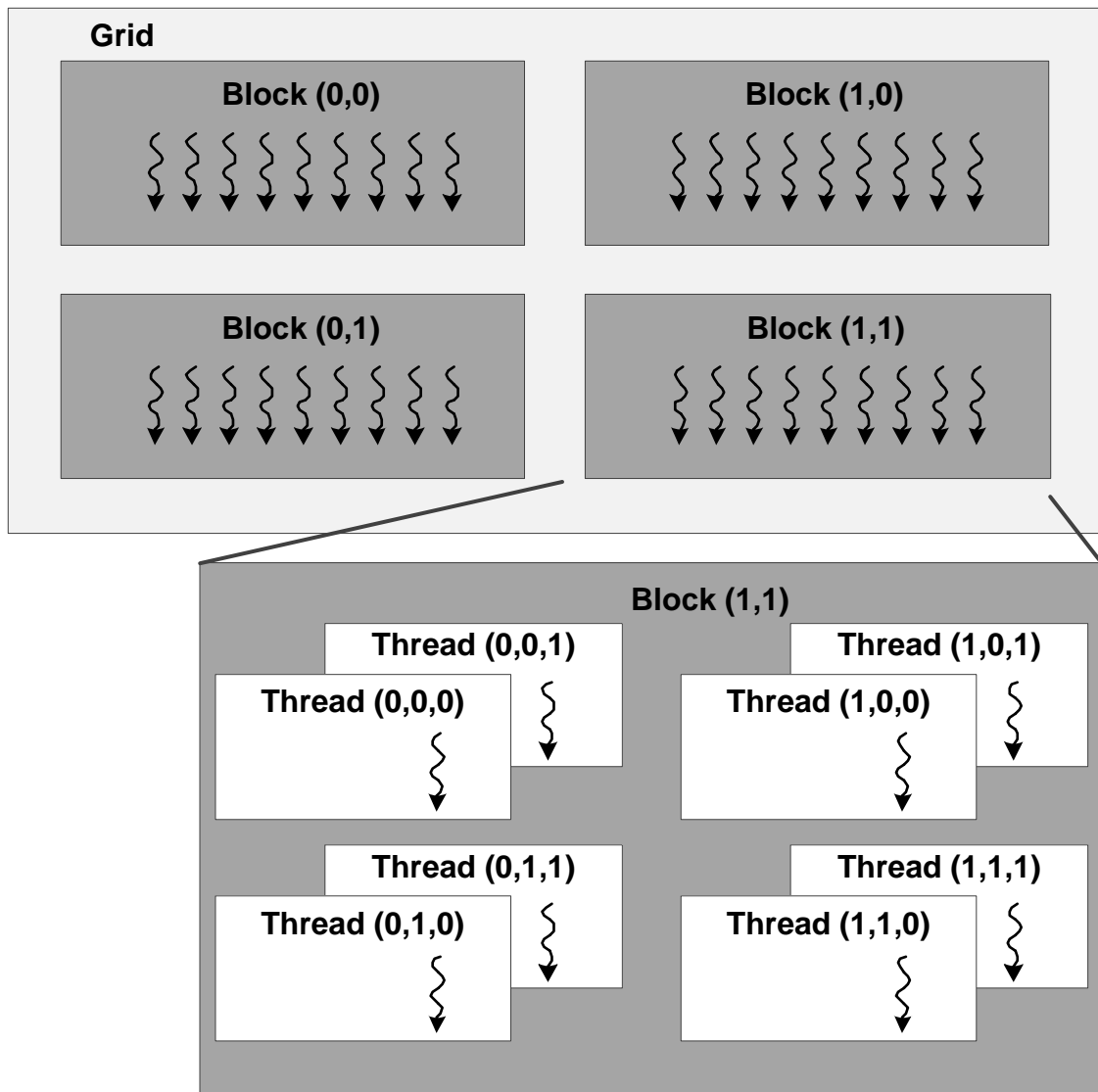


Figure 7.3 Thread hierarchy

For the purposes of explanation, the CUDA code shown in Figure 7.4, illustrates how to add two matrices, A and B, of size $N \times N$ and store the result in matrix C (NVIDIA, 2011).

```

int main()
{
    ...
    // kernel invocation
    dim3 threadsPerBlock (16,16);
    dim3 numBlocks (N / threadsPerBlock.x, N / threadsPerBlock.y);
    MatAdd<<<numBlocks, threadsPerBlock>>>(A, B, C);
    ...
}

// Kernel definition
__global__ void MatAdd(float A[N][N], float B[N][N], float C[N][N])
{
    int i= blockIdx.x*blockDim.x + threadIdx.x;
    int j= blockIdx.y*blockDim.y + threadIdx.y;
    C[i][j]=A[i][j]+B[i][j];
}

```

Figure 7.4 CUDA code sample of matrices addition (NVIDIA, 2011)

The main function in this segment of code is to identify the grid and thread block dimensions. These dimensions are set by execution configuration parameters which are contained within the <<< and >>> brackets. Each block contains 256 threads and is arranged into two-dimensional layout. The third dimension of the thread is considered to be zero by default. The number of thread blocks inside the grid depends on the matrix size. For example if the matrix has 1024 x 1024 dimensions then the grid should contain 64 x 64 blocks. At the end of the main function the kernel `MatAdd` is launched. This kernel contains the code that is executed in a parallel manner.

The `__global__` keyword in front of the kernel declaration indicates that this is a kernel function. Since each generated thread has its own ID, the first two lines in the kernel body code determine the area of a particular part inside the data that a thread needs to work on. For instance, the first generated thread in the first block is assigned to add the first elements in matrix A and B together. In this case, the indices for both thread and block are zeros. That is, the pointers `i` and `j` should have zero values as well.

The CUDA runtime system automatically assigns the generated threads to the GPU cores (Kirk and Hwu, 2010). Even if the number of generated threads is more than the available cores, the thread blocks are scheduled in such a way maintains high parallelism performance.

7.3.2 Memory hierarchy

In CUDA, the host and device have separate memory spaces with different characteristics and performances. These memory spaces are classified as global memory, constant memory, texture memory, local memory, registers, and shared memory as shown in Figure 7.5. This figure shows the types of memories and the thread access directions to them. The main two memory spaces considered in this project are register and global memories.

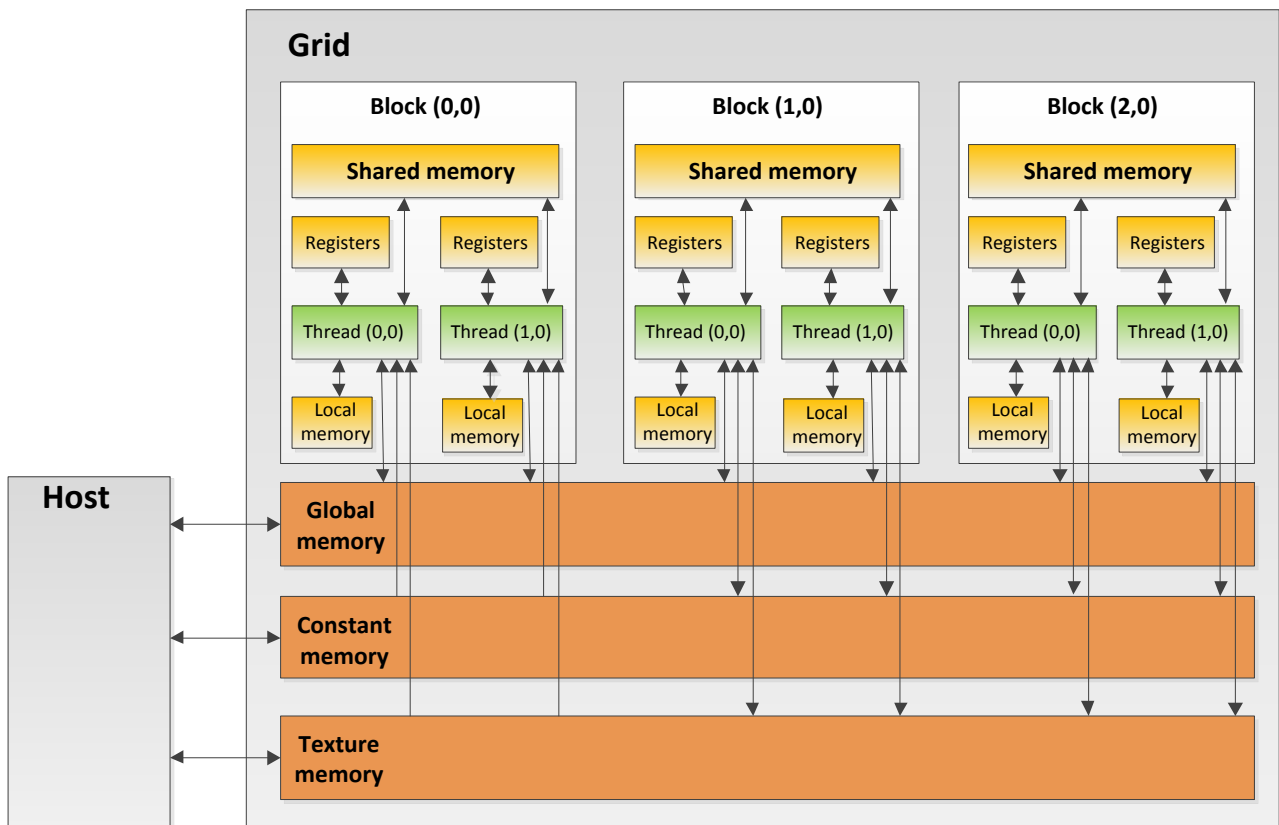


Figure 7.5 Relationships between the memory spaces in device memory and threads

Each thread has private register memories. These registers are on chip memory and as such very fast, with read-write accessibility, but with a limited memory capacity. They are used to store

only frequently accessed variables that are private to each thread, such as thread and block indices (i.e. `threadIdx` and `blockIdx`). The global memory is read-write memory and it is viewable by the entire threads and the host (NVIDIA, 2008).

Since the host can view the global memory, the data that initiated in the CPU can be copied and transferred to this memory which has a large memory space that can store the frames of intensity data captured from the WSI interferometer.

7.4 Acceleration of Computing Process

The evaluation process for areal topography needs a long processing time if traditional CPU sequential execution programs are used. The CUDA C programming construct is used to achieve data parallelism on a GPU, thus increasing the measurement throughput.

This section reports the use of CUDA parallel programming to accelerate the FFT fringe analysis algorithm (algorithm C). This algorithm shows nano-scale accuracy with execution time of 32 seconds when analysing 256 frames. The algorithm steps are summarised as:

1. Remove the envelope function of the background intensity.
2. Apply FFT to obtain the power spectral density.
3. Filter out the unwanted spectral information.
4. Apply inverse FFT to retrieve the phase term.
5. Apply complex logarithm to extract the phase.
6. Unwrap the phase.
7. Fit the phase using linear polynomial least square approximation.
8. Calculate the surface height.

These steps are written as kernels in order to be executed on the GPU except for the second and fourth steps which are executed using FFT parallel functions provided in the NVIDIA® CUDA™ Fast Fourier Transform library (NVIDIA, 2007).

7.5 The Proposed CUDA Structure

The proposed program consists of host code that executes on the CPU and a device code that executes on the GPU as shown in Figure 7.6. The host code is written so as to arrange the captured data in a form suitable for the WSI analysis algorithms described in Chapter 6. After that, the host code launches API CUDA memory arrangement functions to copy the data into the GPU memory spaces. Subsequently, the host declares the configuration of thread organisation. Finally, the host invokes NVIDIA FFT library functions as well as five custom written kernels in sequence to evaluate the data according to the WSI analysis steps described in the previous section.

The invoked device code (kernel) is executed on the GPU by the generated threads. The results of the device code are stored in the GPU memory.

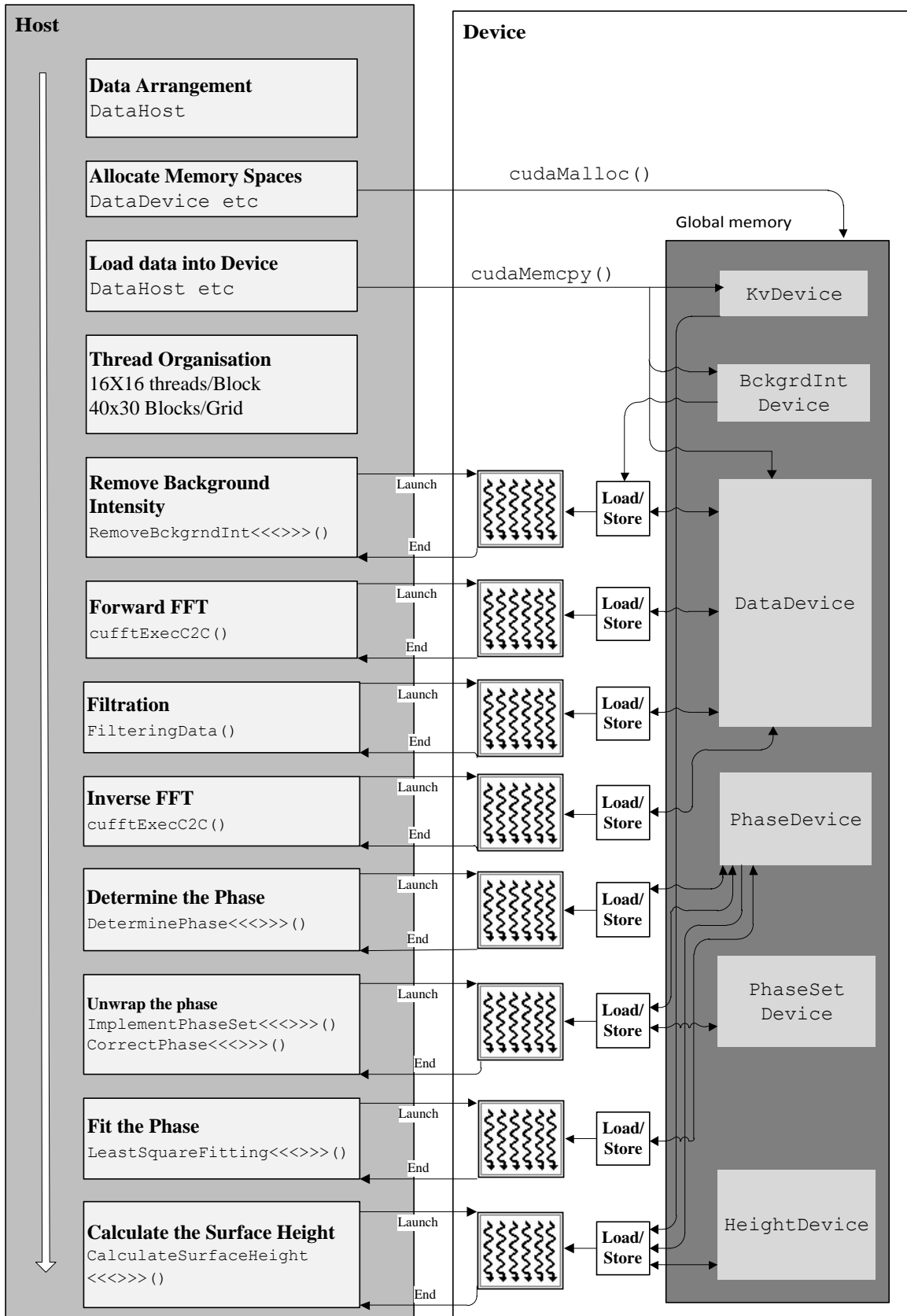


Figure 7.6 The proposed program structure using CUDA

7.5.1 Data Arrangement

First, the host code arranges the data that captured from the CCD. The captured frames are stored in the CPU main memory in a succession as shown in Figure 7.7.

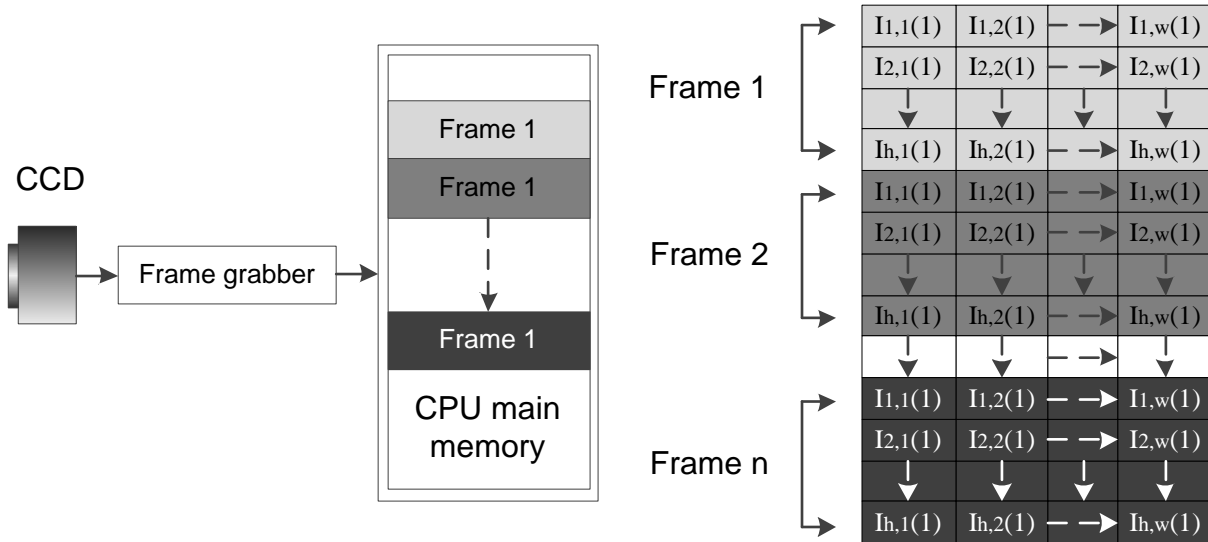


Figure 7.7 The frames order is as obtained from the capturing process. n is the number of captured frames, h is the frame height (e.g. 480) and w is the frame width (e.g. 640)

The first frame is placed in a linear memory followed by the second frame and so on till the last frame. The host code re-arranges this data into a linear array. This array is presented as the variable array `DataHost`. `DataHost` consists of a number of data segments. The number of segments is equal to the number of CCD pixels (640 x 480). The length of each segment is equal to the number of captured frames (e.g. 256). The data segments are placed in succession. That is, the segment of the first pixel is placed at the beginning followed by segment of the second pixel and so on until the segment of the last pixel. The organised form is shown in Figure 7.8. The code that is used to organise the data is shown in Appendix C.1. The full actual code is shown at end of Appendix C.12.

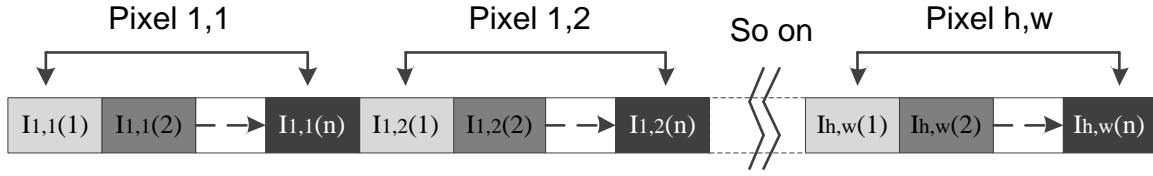


Figure 7.8 Data structure of DataHost

7.5.2 Data Transfer

The data transfer is essential to the CUDA program because the GPU has a separate memory space from the CPU. Thus the essential data captured by the interferometer needs to be transferred to the GPU. These data are the captured frame array, `DataHost` (shown in Figure 7.8), the measured wavenumber, `KvHost` (shown in Figure 5.4) and the calibrated background intensity, `BkgrdIntHost` (shown in Figure 5.9). These arrays are transferred from the CPU main memory to GPU global memory.

The GPU global memory can be viewed by the entire grid. As a result, all the grid threads can access the global memory to read or write data. The transfer process consists of two parts. The first part is to allocate a fraction of the global memory for each array to receive the transferred data. The allocated fractions of memory spaces are presented as `DataDevice` and `KvDevice`. The allocation process is achieved by using the CUDA API function `cudaMalloc()`. The second part is to copy the data from the CPU to the allocated fraction in the device. This is achieved by the CUDA API function `cudaMemcpy()`, see Appendix C.2. The same API functions are used to copy `KvHost` and `BkgrdIntHost` into `KvDevice` and `BkgrdIntHost` respectively.

The communication bandwidth between the GTX280 and the CPU is 12.8 GB/s (Anand Lal Shimpi, 2008). As such, the CUDA application can transfer data from the CPU memory into the global GPU memory at 6.4 GB/s and at the same time uploaded data back at 6.4 GB/s. For example, if the `DataHost` consists of 256 frames, each frame consists of 640×480 pixels, each one having

a float type size of 4 bytes, then the `DataHost` can be transferred from the CPU memory to global GPU memory in approximately 50 ms.

7.5.3 Thread Organisation

The proposed kernels generate a number of threads equal to the total number of CCD pixels. Each thread manipulates the data collected by a corresponding pixel. These threads are organised in a grid which is organised into 40 x 30 arrays of blocks. Each block is organised into 16 x 16 threads, see Figure 7.9. The grid organisation parameters are stored as `struct` variables. The dimension of a grid in terms of blocks is given as `dimGrid` and the dimension of a block in terms of threads is given as `dimBlock` (see Appendix C.3).

This thread hierarchy provides the capability to designate each thread to the offset address of its own data segment, D_s . The designation is achieved by using indices of both blocks and threads as given in

$$D_s = (F_w * N * row) + (N * col) \quad (7.1)$$

where F_w is the frame width, the N is the total number of the captured frames. The *row* and *col* are the row and column positions of the specific threads in the thread hierarchy which can be found by the following equations,

$$row = (B_s * blockIdx.y) + threadIdx.y \quad (7.2)$$

$$col = (B_s * blockIdx.x) + threadIdx.x \quad (7.3)$$

where B_s is the block size in one dimension, which is equal to 16. This parameter can be used in both *row* and *col* calculations because the block has a square hierarchy. For instance, the *row* and *col* values of the first thread in the last block in the proposed program are calculated as,

$$row = (16 * 29) + 0 = 464$$

$$col = (16 * 39) + 0 = 624$$

where the indices of the block and thread are given as (`blockIdx.x=39, blockIdx.y=29`) and (`threadIdx.x=0, threadIdx.y=0`) respectively. Using Equation 7.1 leads to the data segment offset address D_s ,

$$D_s = (640 * 256 * 464) + (256 * 624) = 76181504$$

This address is stored in the register memory of the thread. This concept is applied to all the generated threads. Thus, each thread knows the address of its own data segment.

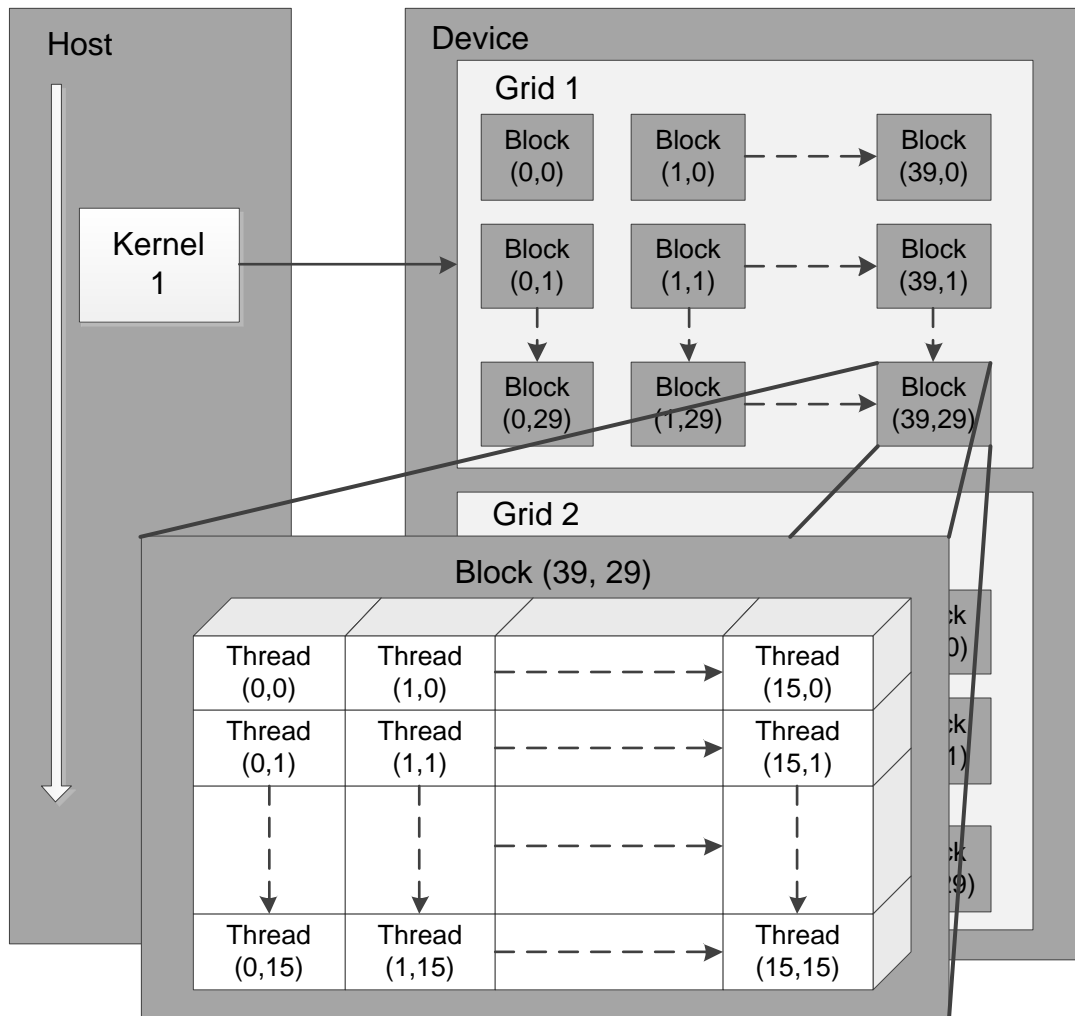


Figure 7.9 Thread organisation

7.5.4 Kernel Invocation and Execution

After the data transaction and thread generation, the NVIDIA® CUDA™ Fast Fourier Transform library functions and six written kernels are invoked in a sequence that presented as a flow chart in Figure 7.6.

The invocation of the kernels are synchronised with GPU execution. Each kernel is invoked from the host after getting a response from the device acknowledging that the execution of a previous operation has been accomplished.

The data processing is started by removing the background intensity from the intensity pattern. This background intensity is removed from the data segments of all pixels in a parallel manner using the `RemoveBckgrndInt<<<>>>()` kernel. This kernel is invoked from the host code, as shown in step 3 of Appendix C.4, and executed on the device. The actual code of the kernel is stated in Appendix C.5. The kernel consists of ten steps. The first six steps in this kernel assign each generated thread to its own data segment using the block and thread indices: `threadIdx.x`, `threadIdx.y`, `blockIdx.x`, and `blockIdx.y`). These indices are stored in the registers of their threads (steps 1 and 2). Then the data segment address of each thread is determined by steps 3 to 6. Finally, steps 7-9 remove the background intensity from the each data segment by dividing the values of each data segment over its measured background intensity in a parallel manner. The last step uses the barrier synchronization function `__syncthreads()` to coordinate thread activity in each block. This step ensures all the threads in a block have completed their kernel execution before moving on to the next task. The manipulated data is stored in the original `DataDevice` array.

Applying FFT using CUDA is based on using the NVIDIA® CUDA™ Fast Fourier Transform library. This operation needs to create a plan in order to store the configuration mechanism calculated by using the CUFFT library. The plan is created by a `cufftPlan1d()` CUFFT library function. The second parameter of this function presents the data segment length which is equal to

the number of frames. The fourth parameter of this function is the number of times it is necessary needs to perform the FFT in parallel. In this program, the fourth parameter represents the number of pixels in the captured frame; $FW \times FH$. The configured plan is handed to the `cufftExecC2C()` function to execute the FFT. The last parameter of `cufftExecC2C()` function decides whether the function should perform forward or inverse FFT. The calculated result is stored in the data array, `DataDevice` as a `float2` type. `DataDevice.x` and `DataDevice.y` are used to store the real and imaginary parts respectively.

The unwanted information is filtered out from all the pixels in a parallel manner by a written kernel called `FilteringData<<<>>>()`. This kernel is invoked from the host code and executed on the device as declared in step 7 of Appendix C.4. The kernel code is listed in Appendix C.6. This kernel structure is similar to `RemoveBckgrndInt` except steps 7-12 which refer to kernel functionality.

The filtered `DataDevice` array is next processed by using the inverse FFT functionality provided by the CUFFT library. To perform the inverse operation the same function is used, i.e. `cufftExecC2C()` with same configuration plan carried out in the forward FFT, but the last parameter is now `CUFFT_INVERSE` rather than `CUFFT_FORWARD`.

Then the inverse FFT of `DataDevice` array is processed with the remaining five kernels (see steps 10-14 in Appendix C.4). Three new arrays are allocated in the device global memory to store and load the results obtained from these kernels, `PhaseDevice`, `PhaseSetDevice` and `Heightdevice` respectively.

The `DeterminePhase<<<>>>()` kernel applies the natural logarithm to `DataDevice` and stores the result in `PhaseDevice`. The `CorrectPhase<<<>>>()` kernel is written to correct discontinuity in the phase by adding `PhaseDevice` to the phase offset distribution obtained from `ImplementPhaseSet<<<>>>()` kernel.

The `LeastSquareFitting<<<>>>()` kernel fits the corrected phase obtained from `PhaseDevice` using linear fitting approximation to remove the non-linear deviations existed at the edges of the phase distribution.

The last kernel `CalculateSurfaceHeight<<<>>>()` is written to calculate the surface height for areal topography from the `PhaseDevice` array and store the result in the `HeightDdevice` array. These five kernels were written in the same manner as the `RemoveBckgrndIntkernel`, see Appendix C. 7-10).

Finally, the calculated surface height data `HeightDdevice` is again copied to the host to be written into the SDF file to be viewed by commercial software packages such as `SurfStand`. All the arrays created within the device memory are erased using the `cudaFree()` CUDA API function.

7.6 Surface Representation

There are many commercial software packages used for analysing surface data and visualising the data graphically, for example `Surfstand`. In order to use these software packages the calculated surface height data is written into a universal Surface Data File (SDF) format (see Appendix C.11).

The format of the SDF file is divided into header, data area and trailer as stated in table 7.2 (NIST, 2002). The header type contains general information about the measurement system, measurement date, the measurement lateral and vertical scales etc. The total length of the header is 81 bytes.

The data area contains the surface height values follows the header. These values are scaled by the Z-scale factor that exists in the header. The trailer section of the SDF file contains historical information such as the operator name and measurement conditions which are ignored in this research.

Table 7.1 Format of the created SDF file

File Section	Information	ASCII Name	Binary Data Type	Binary Length (Bytes)
Header	Version number	VerName	CHAR	8
	Manufacturer's ID	ManufacID	CHAR	10
	Creation Date and Time	CreateDate	CHAR	12
	Last Modification Date and Time	ModDate	CHAR	12
	Number of Points per Profile	NumPoints	UNSIGNED INT	2
	Number of Profiles	NumProfiles	UNSIGNED INT	2
	X-Scale	XScale	DOUBLE	8
	Y-Scale	YScale	DOUBLE	8
	Z-Scale	ZScale	DOUBLE	8
	Z-Resolution	Zres	DOUBLE	8
	Compression Type	Compres	DOUBLE	1
	Data Type	DataType	CHAR	1
	CkeckSum Type	CkeckType	CHAR	1
Data area	SurfaceHeightHost		FLOAT	1.2288M
Trailer	NULL	NULL	NULL	NULL

7.7 Result and Discussion

The effectiveness of parallel programming with the proposed system has been investigated by measuring the execution time of four captured frames sets: 64, 128, 256, and 512. These frame sets represent the captured data for the 4.707 μm step height sample. The data has fixed captured frame size of 640 x 480 pixels. The execution time and accuracy obtained from CUDA program, using GeForce GTX 285, are compared to those obtained using a sequential Matlab implementation on a

quad-core AMD Phenom™ II processor clocked at 2.5 GHz. Both programs results are stored in SDF format and analysed using the SurfStand software package.

The results of acceleration are shown in Table 7.2. These results demonstrate that the execution time can be reduced to the order of a few seconds or less if a CUDA-GPU is used to evaluate the surface height. For instance, if the number of frames required to be captured in the WSI system is 256 frames the overall capture time is approximately 2.6 seconds and the computing time is approximately 1.2 second. Consequently the WSI measurement time is 3.8 second. This time is reduced to about 1.8 second if 128 frames are used for the measurement.

However, the acceleration factor (ratio of processing times) decreases as the captured frame number increases. Typically this is because of the memory limitation of the GTX 285 in the proposed measurement method. This reduction in the acceleration factor can be improved by using up-to-date GPUs which are continuously improving in terms of both number of processing cores and memory.

Table 7.2 Parallel programming performance versus sequential programming

Frame Set	64	128	256	512
Matlab Processing Time (msec)	12037	20712	31990	61223
CUDA C Processing time (msec)	182	422	1159	2845
Acceleration Factor	66.1	49.1	27.6	21.5

The effect of the computer word length has been investigated, since the floating point arithmetic units of the GTX 258 are single precision while the CPU features double-precision. The accuracy of the results was examined by calculating the absolute maximum difference of several points obtained from the CUDA and Matlab implementations.

Table 7.3 shows that the absolute differences are very much less than a nanometre and this satisfies the requirements of the WSI system.

Table 7.3 Accuracy of CUDA program with respect to Matlab

Frame Set	64	128	256	512
Absolute maximum difference (nm)	0.00312	0.00626	0.00295	0.00435

7.8 Summary

The combination of CUDA and a GPU can be used to accelerate the execution time of arithmetic calculation with floating point operation. The acceleration factor depends on the GPU hardware resources, specifically the number of cores and memory capacity. For 256 frames, the execution time is seen to be reduced from approximately from 32 seconds to 1.2 seconds. This acceleration can effectively reduce the measurement throughput.

The accuracy of single floating point operation satisfies the WSI nanoscale measurement requirement where it is found to be better than 0.007 nm.

The parallel programming model can generate many thousands of threads to execute the code in a parallel manner. Each generated thread is assigned to a specific data segment. The many cores and the memory hierarchy of the GPU, such as the global and register memories, give the threads the ability to achieve high computing performance.

As a result, this cost effective solution enables high computing performance for the WSI system.

8. Compensation for Environmental Disturbances

8.1 Introduction

Environmental disturbances such as vibration and temperature gradients can cause unwanted changes in the optical path difference (OPD) of interferometers which results in measurement error, if they are greater than the measurement resolution.

This chapter introduces a method to stabilise the proposed WSI system against mechanical disturbances. The disturbances are controlled actively by using a piezoelectric translator (PZT) attached to the reference mirror. This method can compensate for a disturbance in the OPD up to a few micrometres and several hundreds of Hz. Since most building vibration occurs in the region of 20-200 Hz (Hayes, 2002, Leach, 2010), this compensated WSI can be valuable for use online/in-process on the shop floor.

This chapter starts by introducing the method of stabilising the WSI. A mathematical description using transfer functions and block diagrams is presented in this chapter. The system is simulated using Matlab Simulink to theoretically investigate the stabilisation effectiveness. Finally the compensated WSI is tested practically by introducing a mechanical disturbance at range of frequencies into the measured sample. The results are also presented for both theoretical and practical tests.

8.2 System Implementation

The compensated system was implemented by multiplexing a reference interferometer with the main WSI that introduced in Chapter 5 (see Figure 8.1). The symbols on the figure have been previously defined in Chapter 5. The prototype instrument is shown in Appendix A.2. The components of this system consist of:

- Wavelength scanning interferometer (WSI) as described in Chapter 5.
- Super-luminescent light emitting diode (SLED) with a centre wavelength of 816 nm and optical power of 7 mW (Exalos EXS8310-B001).
- Two edge-pass filters (EP1 and EP2). These are di-chroic mirrors have a cut-off wavelength at 800 nm (Thorlabs FES0800). Multimode optical fibre (OF) having a 600 μm core and a transmission range from visible to near infrared (Thorlabs FT600EMT). Two lenses (COL and CL) having a focal length of 30 mm.
- A silicon amplified detector (PD) having a maximum responsivity at 800 nm (Thorlabs PDA8A). A servo electronic circuit which consists of a Proportional-Integral (PI) controller and amplifiers (see Appendix D.1). A piezoelectric translator having a maximum travel range of 15 μm (Physikinstrumente PI-840-10).

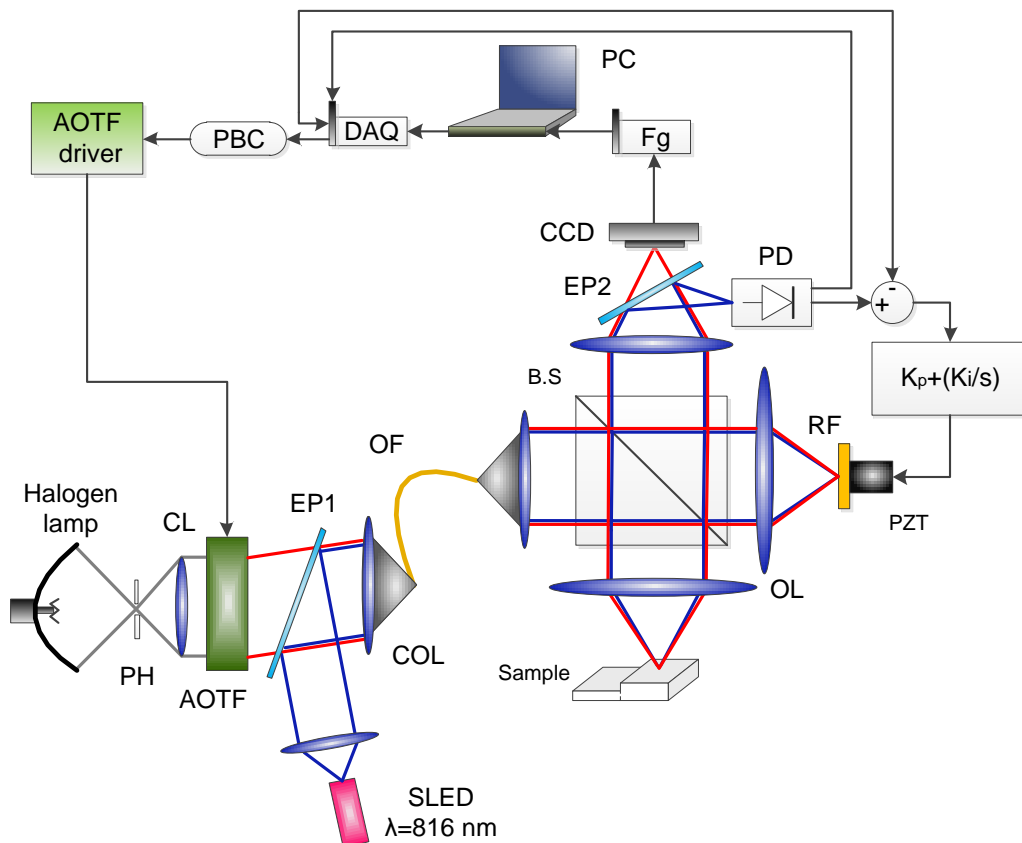


Figure 8.1 Configuration of the proposed compensated interferometer

The reference interferometer is illuminated by a super-luminescent light emitting diode (SLED). This SLED operates at a wavelength far from the scanned wavelength region of the WSI interferometer. The peak wavelength of the SLED is 816 nm while the wavelength scanning range of the WSI is 683.42 nm - 590.98 nm.

The two beam paths of the interferometers are combined by edge-pass filter EP1. EP1 has a cut-off wavelength at 800 nm. Thus the beam diffracted from the AOTF is passed through EP1 while the beam emitted from the SLED is reflected. The two beams are combined together and coupled into a multimode optical fibre by a coupling lens, COL. The beams emerge from the fibre and are collimated by lens CL into a Linnik interferometer. The two beams therefore share a common optical path through the interferometer. This effectively creates a reference and measurement interferometers sourced by the SLED and filtered halogen source respectively. As a result of the shared optical path, the environmental noise in the measurement interferometer is compensated when the reference interferometer is stabilised.

The wavelengths of the two interferometers are separated by another edge-pass filter, EP2. The WSI interferometer output is detected by a CCD camera, while the reference interferometer output is detected by an amplified silicon photo-detector (PD). The detected interference signal of the reference interferometer is then fed back to a PI controller. The output control signal of the PI controller drives a PZT which is attached to the reference mirror of the multiplexed interferometer. As a result, any disturbance in the optical path difference is compensated by the PZT. The PD signal also feeds the DAQ card to measure the visibility which is important in order to remove the DC offset component from the interference signal.

This chapter describes the compensation mechanism of the multiplexed interferometer and following sections give a block diagram representation of the reference interferometer.

8.3 Block Diagram of the Reference Interferometer

The reference interferometer $G_P(s)$ can be considered as the plant in a closed loop system. This plant is controlled by a PI controller $G_C(s)$ and a PZT $G_{PZT}(s)$. The output of the plant $C(s)$ is fed back to a summing point, where it is compared to a desired value known as a reference input $R(s)$. The error signal, $E(s)$ is defined as the difference between $C(s)$ and $R(s)$. The controller evaluates the error signal, $E(s)$ and produces a control action to eliminate the error signal. Figure 8.2 illustrates the block diagram of the closed loop system.

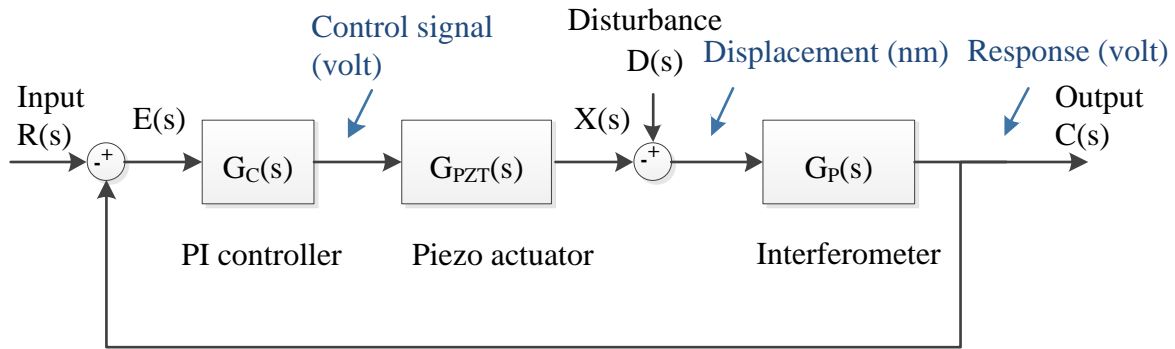


Figure 8.2 Block diagram of the closed loop system. The $G_P(s)$, $G_C(s)$ and $G_{PZT}(s)$ are transfer functions of interferometer, controller and the PZT respectively

The reference interferometer here acts as displacement sensor and by virtue of its common optical path, it is to be subjected to the same disturbance as the WSI. If the reference input, $R(s)$ is held at zero, then $E(s)$ is related only to the disturbance caused by environmental effects. As such, the compensation is limited to the reducing that disturbance only. The interferometer is characterised using the Simulink trigonometric function to simulate the cosine term of the interferometer response. The transfer function of each block in Figure 8.2 will be discussed below.

The output variable of each block must be matched with the input variable of the next successive block. Consequently, the electrical control signal produced by the controller needs to be converted into a form which compensates any displacement disturbances that occur in the

interferometer i.e. it must convert voltage into displacement. It is the PZT that transforms the electrical signal into displacement. The PZT mechanical movement follows the displacement of the disturbance.

This operation should keep the OPD fixed resulting in a stable fringe pattern being obtained. However, if any fluctuation occurs in the OPD due to insufficient control action, the reference interferometer converts this fluctuation displacement into an electrical signal which is fed back to the controller. This electrical signal is generated by detecting the fluctuating fringes due to vibration via photo-detector PD. The controller then re-evaluates the error signal and updates the control action. The following section describes each element individually and presents its transfer function.

8.3.1 Controller representation

The vital part in the closed loop system is the controller that drives the PZT to stabilise the compensated system. The controller is a PI type which is characterised by the transfer function (Ogata, 2002),

$$G_c = K_p + \frac{1}{T_i s} \quad (8.1)$$

where K_p is the proportional gain and T_i is the integrating time. The parameters K_p and T_i are tuned manually by trial and error. It was found that with $K_p=0.5$ and $T_i=86.36 \mu\text{sec}$ stabilisation can be achieved. Thus the PI controller can be represented by the transfer function,

$$G_c = 0.5 + \frac{1}{86.36 \times 10^{-6} s} \quad (8.2)$$

This yields to a block diagram shown in Figure 8.3.

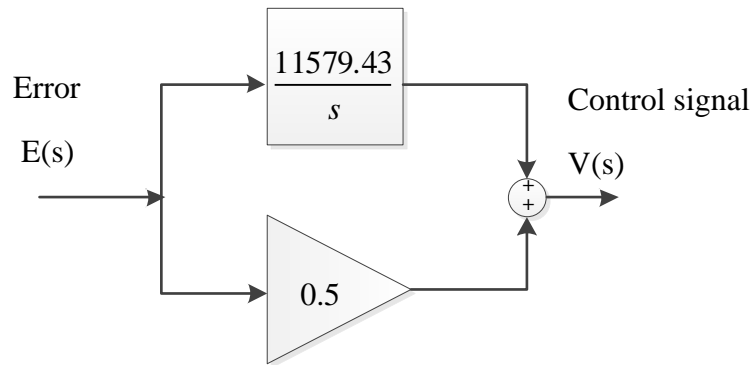


Figure 8.3 Block diagram of the controller

8.3.2 PZT transfer function

This section introduces the transfer function for the PZT. This mathematical approach is for a ceramic PZT multilayer stack actuator, a simple model is shown in Figure 8.4. The symbols used are listed Table 8.1.

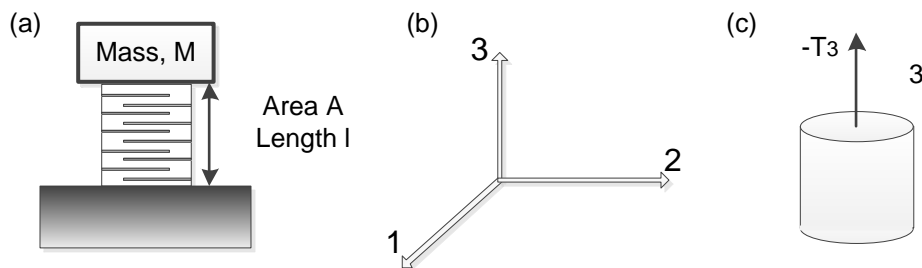


Figure 8.4 Simple PZT and mass model (b) directions of forces affects a PZT element (c) PZT model response in direction 3 only

Table 8.1 Glossary of symbols

Symbol	Unit	Description
X	m	Displacement in a piezoelectric by an applied field, E
E_3	V/m	Electric field (voltage per unit distance between electrodes) in direction 3.
m	kg	Mass of the actuator.
d_{33}	m/V	Piezoelectric strain coefficient (i.e. induced strain in direction 3 per unit electric field applied in direction 3). (also known as the charge constant)

l	m	Length of piezoelectric.
A	m	Cross-sectional area.
s_{33}^E	m^2/N	Elastic compliance
c	N/m^2	Modulus of elasticity (Young's modulus).
S_3	No unit	Strain of piezo in direction 3.
T_3	N/m^2	Stress of piezo

The force that is produced by the PZT can be represented by:

$$F = -T_3 A \quad (8.3)$$

The fundamental PZT equation below, for the longitudinal mode, describes the strain of the PZT element in terms of material properties and the stress generated by the electric field (Uchino and Giniewicz, 2003, Giurgiutiu and Lyshevski, 2004).

$$S_3 = s_{33}^E T_3 + d_{33} E_3 \quad (8.4)$$

Since the type of PZT which is used in this project, consists of a stack of ceramic disks separated by thin metallic electrodes, the displacement of the piezo actuator is related to the number of ceramics layers, in addition to the applied voltage and material properties (Physik, 2009). Therefore, the strain coefficient d_{33} is multiplied by the number of the ceramics layers n and Equation 8.4 is be modified,

$$S_3 = s_{33}^E T_3 + n d_{33} E_3 \quad (8.5)$$

yielding,

$$T_3 = \frac{S_3 - n d_{33} E_3}{s_{33}^E} \quad (8.6)$$

Substituting Equation 8.4 in Equation 8.6 gives

$$F = \frac{A}{s_{33}^E} (n d_{33} E_3 - S_3) \quad (8.7)$$

To derive a PZT equation that includes the deformation, x caused by the applied electric field, Newton's second law ($F = m \cdot d^2x/dt^2$) is substituted into Equation 8.7 (Lin et al., 1992) to obtain,

$$m \cdot \frac{d^2x}{dt^2} = \frac{A}{s_{33}^E} (n d_{33} E_3 - S_3) \quad (8.8)$$

The strain, S_3 can be replaced by (x/l) . For a longitudinal mode, the elastic compliance s_{33}^E is the reciprocal of the modulus of elasticity, c (APC International, 2011). By adding the damping effect the PZT equation becomes,

$$m \frac{d^2x}{dt^2} + \beta \frac{dx}{dt} + \frac{Ac}{l} x = (A n d_{33} c) E_3 \quad (8.9)$$

where β represents the damping coefficient. The Laplace transformation of Equation 8.9 is,

$$m s^2 X(s) + \beta s X(s) + \left(\frac{Ac}{l}\right) X(s) = A n d_{33} c E(s) \quad (8.10)$$

This yields the PZT transfer function,

$$\frac{X(s)}{E(s)} = \frac{\frac{A c n d_{33}}{m}}{s^2 + \frac{\beta}{m} s + \frac{Ac}{m l}} \quad (8.11)$$

Since $E \cdot l = V$, the transfer function can be re-written in the form of displacement and applied voltage,

$$\frac{X(s)}{V(s)} = \frac{\frac{A c n d_{33}}{m l}}{s^2 + \frac{\beta}{m} s + \frac{Ac}{m l}} \quad (8.12)$$

The PZT transfer function can also be represented by a mechanical spring-damper system attached to a mass i.e. that of the reference mirror, as shown in Figure 8.5.

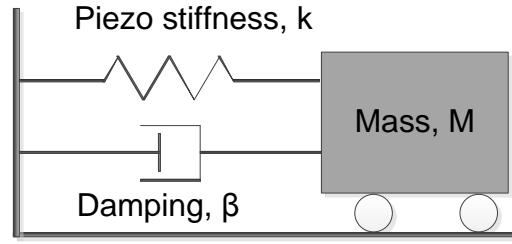


Figure 8.5 Mechanical spring-damper system

The general characteristic equation of figure 8.5 is,

$$s^2 + \omega_n / Q_m s + \omega_n^2 = 0 \quad (8.13)$$

where ω_n is the natural angular frequency of the translator and the Q is the mechanical quality factor. The stiffness of the PZT and its natural frequency can be obtained by comparing the characteristic Equation 8.13 to the denominator of Equation 8.12 as,

$$k = \frac{Ac}{l} \quad (8.14)$$

$$\omega_n = \sqrt{\frac{k}{m}} \quad (8.15)$$

Substituting Equations 8.14 and 8.15 into equation 8.12 leads to,

$$G_2(s) = \frac{X(s)}{V(s)} = \frac{\omega_n^2 n d_{33}}{s^2 + \omega_n / Q_m s + \omega_n^2} \quad (8.16)$$

The response of the PZT can be simulated by determining the parameters in Equation 8.16. The number of ceramics layers and the piezo strain coefficient can be obtained from the piezo data sheet. The natural frequency of the PZT and its quality factor are determined in the following sections.

The natural angular frequency ω_n

The PZT datasheet shows that the static stiffness and the resonance frequency are 57 N/ μm and 18 kHz respectively (PI Ceramic, 2010). As such, the unloaded actuator mass (m) is found to be 4.456×10^{-3} kg using equation 8.15. However, the PZT is loaded by the reference mirror of the interferometer, so the mass is increased by the mass of this mirror. This additional mass will shift the natural frequency according to Equation 8.15. The mass of the attached mirror was measured using a Mettler Toledo lab balance and found to be 935 mg. Thus the total effective mass (m_{eff}) is,

$$\begin{aligned} m_{\text{eff}} &= m + \text{additional mass} \\ &= 4.456 \times 10^{-3} \text{ kg} + 0.935 \times 10^{-3} \text{ kg} \\ &= 5.391 \times 10^{-3} \text{ kg} \end{aligned} \tag{8.17}$$

From equation 8.15, the additional mass of the mirror has lowered the resonance frequency to $f_n = 16.3653$ kHz.

The Quality factor Q_m

The quality factor, Q_m characterizes the sharpness of the resonance. The quality factor for a given PZT is unknown. However, it can be calculated by determining the damping factor, ζ because $Q_m = 1/2\zeta$.

The damping factor is determined by measuring the response PZT to a step excitation. An experiment is carried out to measure the response using a Linnik interferometer. The PZT is attached to a reference mirror of the interferometer as shown in Figure 8.6. The fringe pattern of the interferometer is projected onto the detector.

The PZT movement is analysed by monitoring the intensity fluctuation of the fringe on the detector. Since the intensity fluctuation has a sinusoidal relationship with the phase change due to the cosine term, the interferometer arms are adjusted around the quadrature point of the fringe

pattern (where the cosine term at $\pi/2$). Thus, an approximate linear relationship between the phase change and intensity fluctuation can be obtained.

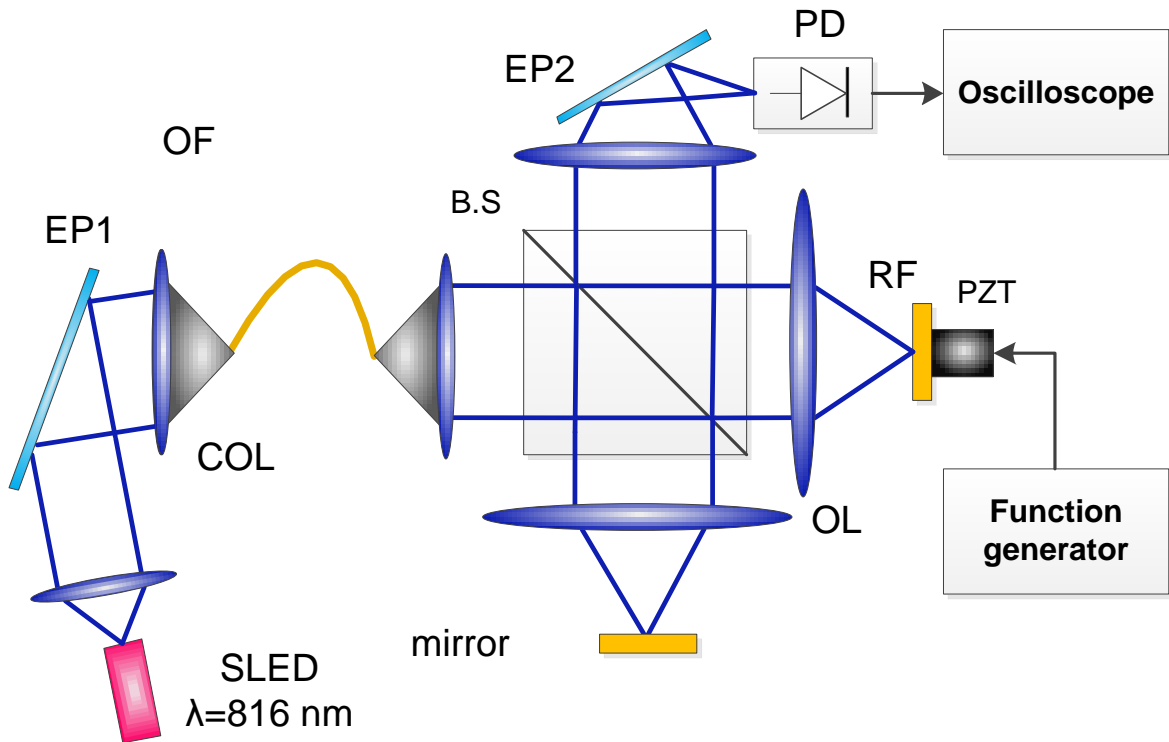


Figure 8.6 Open loop step response experimental setup

Finally a step signal of 0.3V was applied to the PZT in order to move the reference mirror 45 nm (PI-Ceramic, 2010a). This amount of movement ensures the operating region to be approximately linear (see Figure 6.7).

The voltage output from the detector (represents the interferometer intensity) is displayed on an oscilloscope as shown in Figure 8.8. This figure shows the overshoot that occurs when applying the step signal. The damping ratio, ζ , is calculated by measuring the overshoot. The voltage values of maximum peak and steady state are 160 mV and 110 mV respectively. Hence the overshoot percentage is approximately 45%.

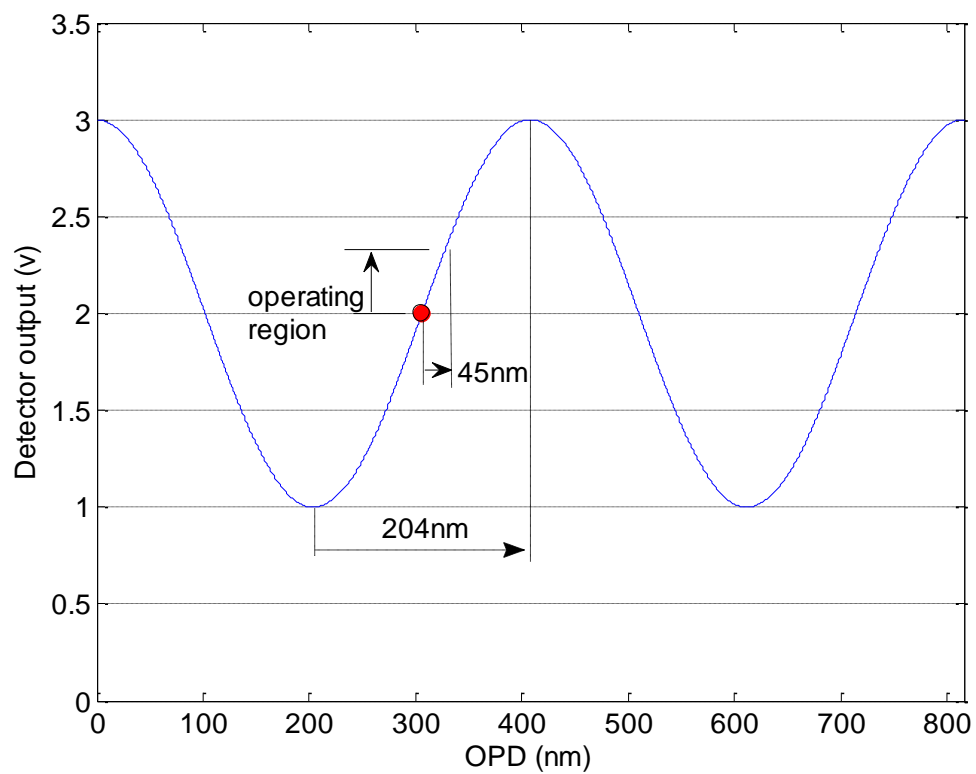


Figure 8.7 Operating region of piezo step response

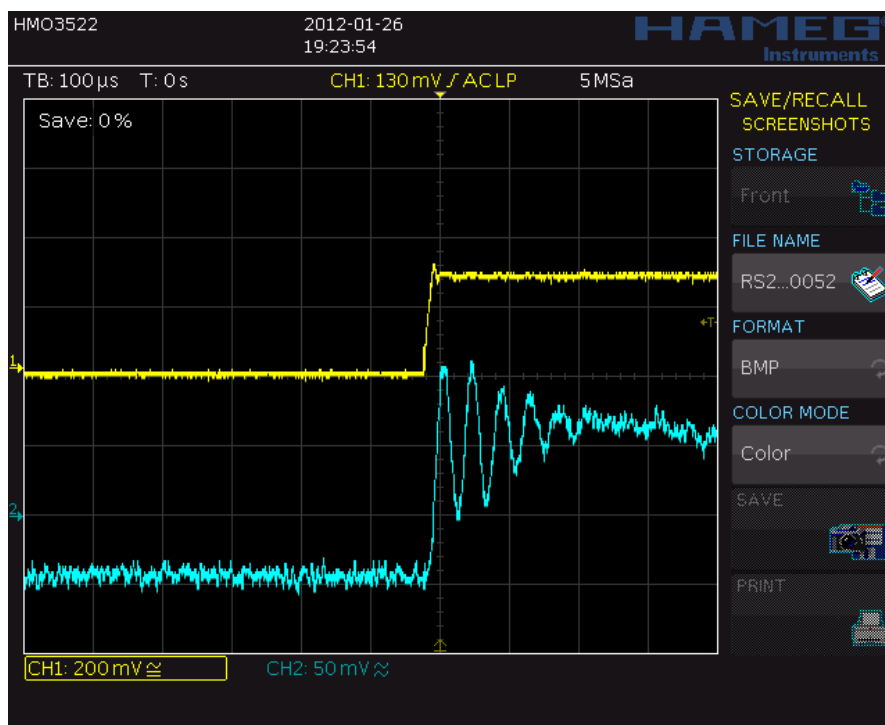


Figure 8.8 Step response of the PZT

The damping ratio can be calculated using the following equation (Ogata, 2002)

$$M_p = e^{-(\xi/\sqrt{1-\xi^2})\pi} \quad (8.18)$$

The overshoot amplitude M_p is approximately 0.45. Thus,

$$\ln(0.45) = \ln(e^{-(\xi/\sqrt{1-\xi^2})\pi})$$

$$0.79 = \frac{\xi\pi}{\sqrt{1-\xi^2}}$$

The damping ratio was found to be $\xi=0.24$. Since the $Q_m=1/2\xi$, the quality factor is approximately equal to $Q_m=2$ in this system.

The transfer function of the PZT can be determined by substituting values of the quality factor ($Q_m=2$) and natural frequency ($f_n=16.3653\text{kHz}$), the number of ceramics layers ($n=375$), and the strain coefficient ($d_{33}=400*10^{-12} \text{ m/V}$) into equation 8.16. The values of n and d_{33} are obtained from the PZT data sheet (PI-Ceramic, 2010b, **Physik**, 2009).

$$\frac{X(s)}{V(s)} = \frac{(2 * \pi * 16.3653 * 10^3)^2 * 400 * 10^{-12} * 375}{s^2 + (2 * \pi * 16.3653 * 10^3 / 2)s + (2 * \pi * 16.3653 * 10^3)^2}$$

Thus the numerical transfer function of PZT (PI-840-10) attached to a mirror has mass 0.935 g is illustrated in Figure 8.9 and given in the following equation:

$$G_{PZT}(s) = \frac{1586}{s^2 + 51413 s + 10573229978} \quad (8.20)$$

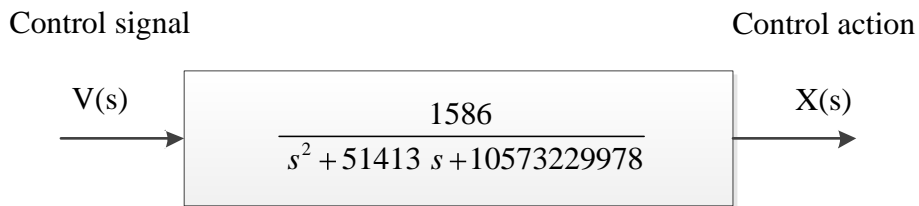


Figure 8.9 Block diagram of the PZT

8.3.3 Interferometer Block Diagram

As discussed in Chapter 3, the interference signal of a Linnik interferometer consists of a DC bias term and a cosine term multiplied by its visibility. By assuming the beam intensity in both arms are equal, the interference signal can be represented in the form,

$$I = 2I (1 + v \cos \varphi) \quad (8.21)$$

and

$$\varphi = \frac{4\pi}{\lambda} [h \mp (d \mp x)] \quad (8.22)$$

where the v is the fringe visibility, h is the surface height which needs to be measured, d is the disturbance displacement and x is the displacement generated by the PZT due to the control action. The above mathematical description for the interference signal can be presented as a block diagram as shown in Figure 8.10.

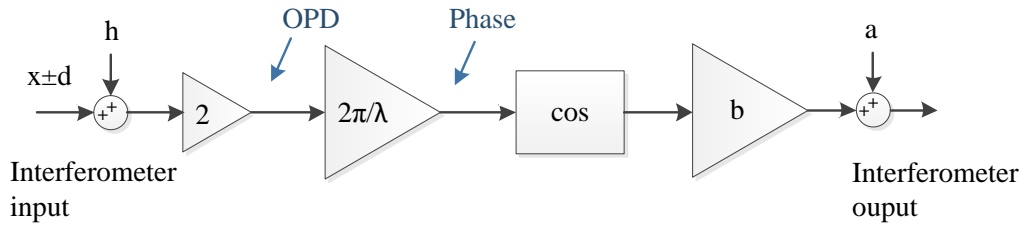


Figure 8.10 Block diagram of the interferometer

where a is the intensity bias which is equal to $2I$ and b is equal to $2I \cdot v$. The fringe visibility, v is measured by modulating the fringes cast upon the detector. The maximum and minimum intensities are found to be 4.2V and 1.8V respectively (see Figure 8.11).

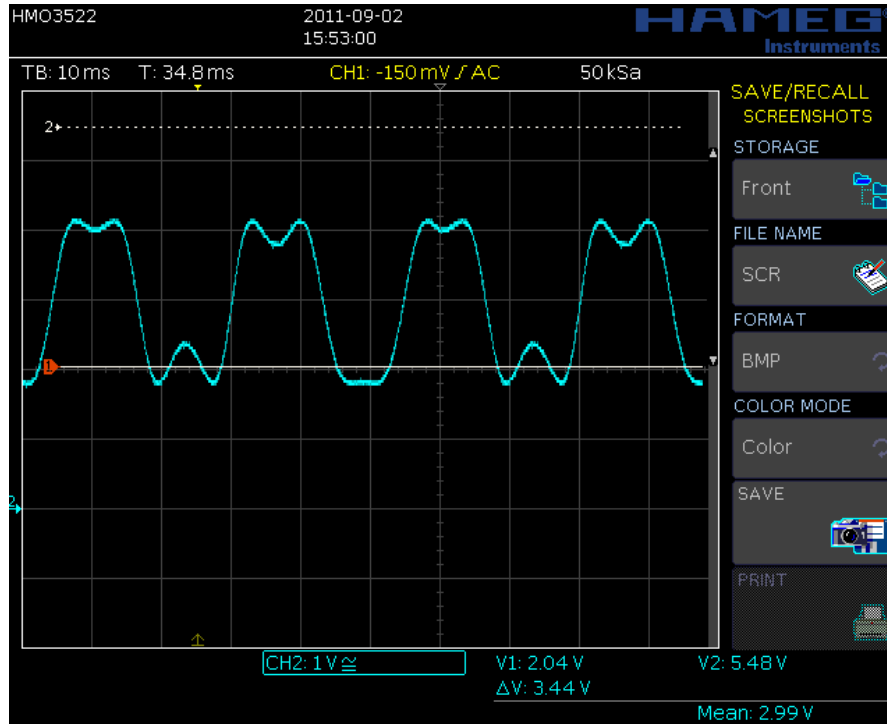


Figure 8.11 Fringe visibility

The fringe visibility is determined by using the visibility equation,

$$v = \frac{I_{\max} - I_{\min}}{I_{\max} + I_{\min}} \quad (8.23)$$

$$v = \frac{4.2 - 1.8}{4.2 + 1.8} = 0.4$$

Furthermore, a is found to be $2I = 2.99V$. This yields the interference signal,

$$I = 2.99 + 1.196 \cos \varphi \quad (8.24)$$

or

$$I \cong 3 + 1.2 \cos\left(\frac{4 * \pi}{816 * 10^{-9}} * OPD\right) \quad (8.25)$$

The block diagram of the interferometer is given in figure 8.12 which can be simulated using Simulink.

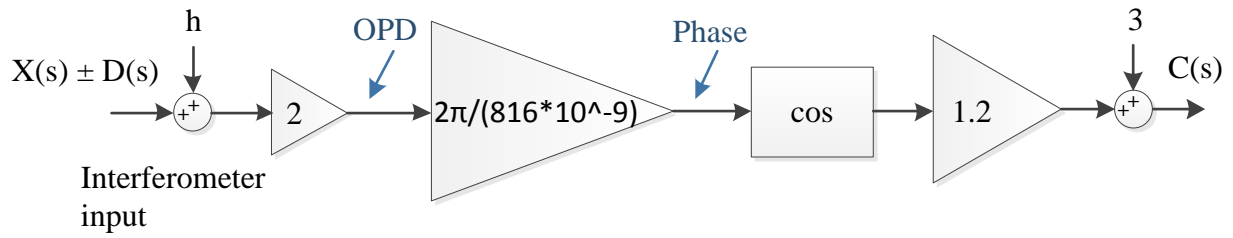


Figure 8.12 Quantified block diagram of the interferometer

8.4 Simulation of Close Loop Compensated System

The block diagram of Figure 8.2 was simulated using Simulink. The transfer functions of the PI controller and the PZT, derived in Sections 8.3.1 and 8.3.2, were used in the simulation. The reference interferometer, which functioned as displacement sensor, is represented in the simulation by using the Simulink trigonometric function.

One of the common vibrational disturbances is the periodic form, hence a continuous sinusoidal form is considered. A step response is considered to simulate a sudden applied force. As such, these forms of disturbances were applied to the simulated block diagram using a virtual function generator. The system response is acquired from the point C(s) using a virtual oscilloscope (see Figures 8.13, 8.14). These results show that the settle time of the compensation is approximately 0.3msec for different types of disturbances. The sinusoidal disturbance amplitude and frequency were set to 0.4 μ m and 40Hz respectively. The step final value was set to 0.4 μ m. The simulation diagram is shown in Figure 8.15.

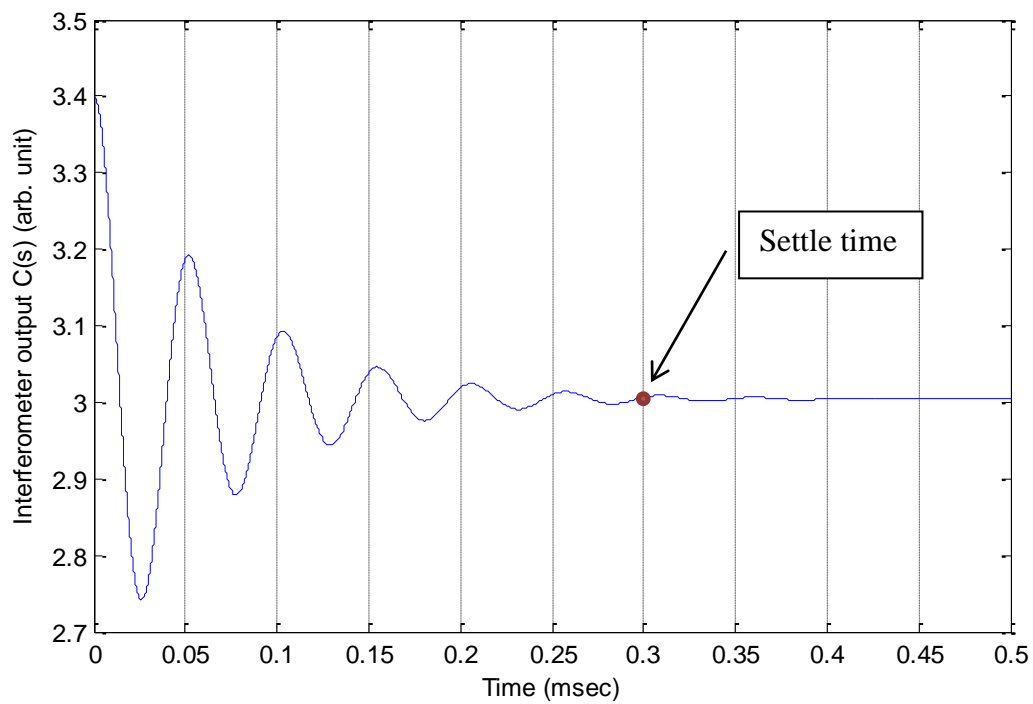


Figure 8.13 Sine response

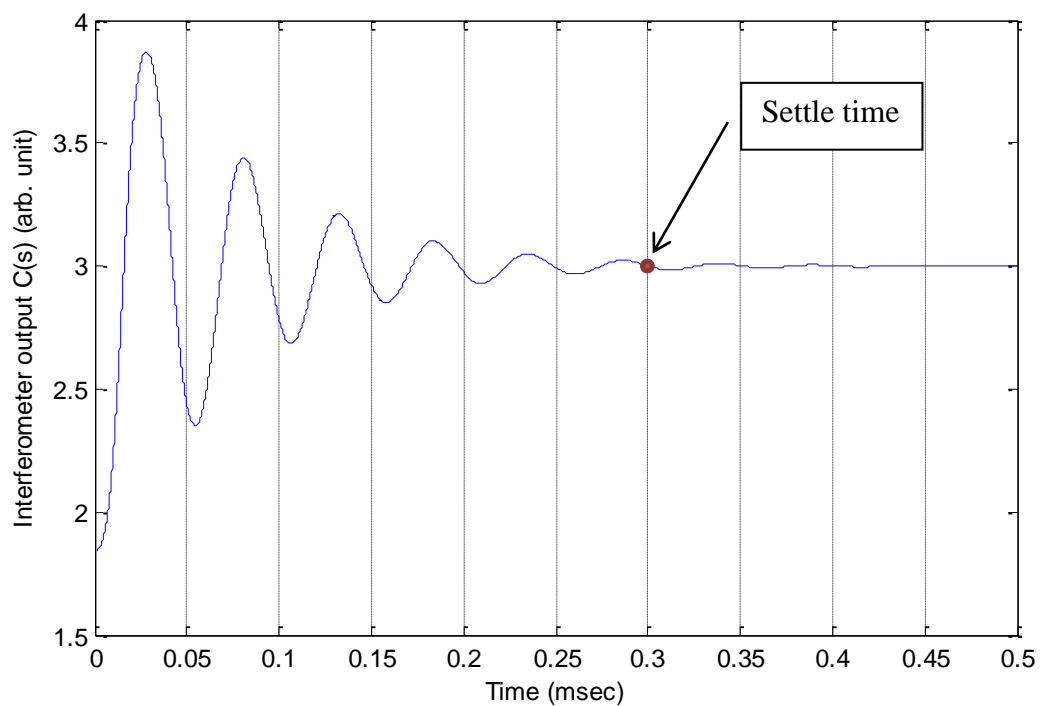


Figure 8.14 Step response

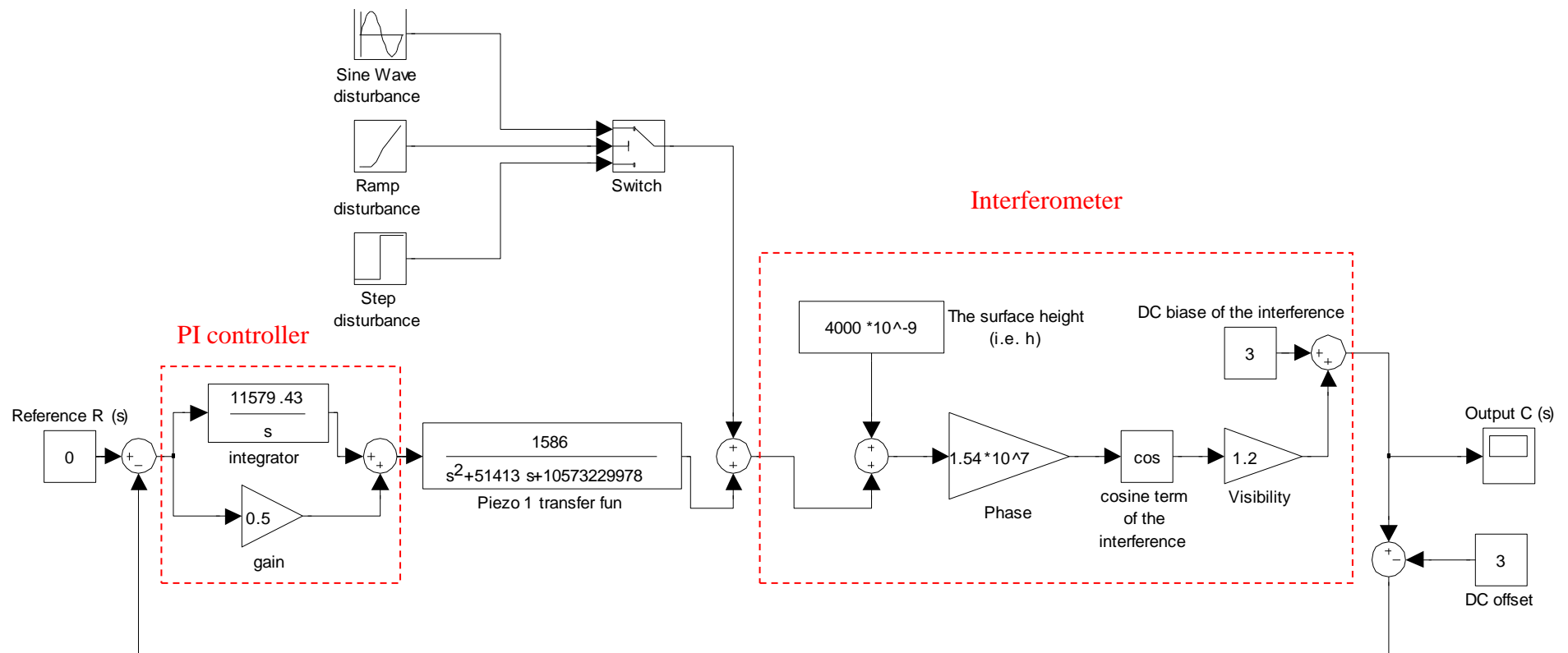


Figure 8.15 Close loop simulation

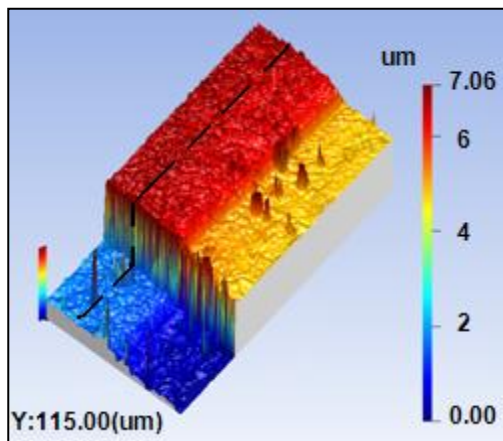
8.5 Stabilisation Practical Performance and Results

The effectiveness of the vibration compensation, as discussed in Section 8.4, was investigated practically by integrating the stabilisation into the WSI interferometer, section 8.2, and the controller was implemented electronically (see Appendix D.1 and D.2).

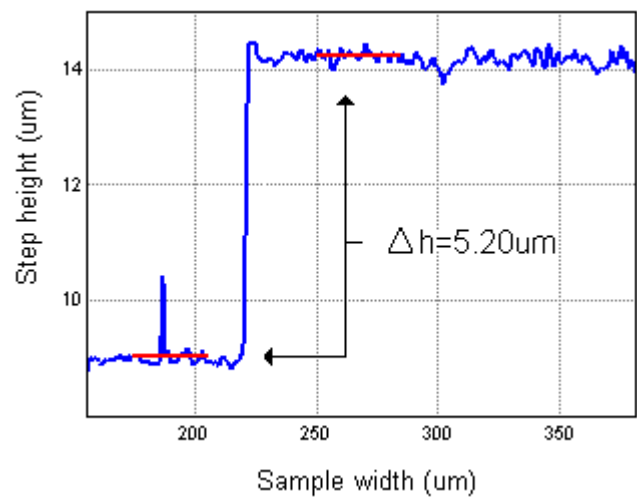
A multi-step optical chip sample was measured before and after stabilisation while being subjected to a mechanical disturbance. The disturbance is a sinusoidal mechanical oscillation of 40Hz and approximately 323 nm peak to peak. The investigation consisted of three parts. The first part was to measure the sample without any mechanical disturbance to obtain a reference measurement, see Figure 8.16(a). The cross-section plot of the sample, seen in figure 8.16 (b), shows one of the steps with height equal to 5.20 μm .

The second part was when the mechanical disturbance was applied using a second PZT attached to the base of the sample. During the disturbance, the sample surface was measured, see Figure 8.17, which shows a result that is so noisy that the step height of the sample cannot be recognised.

Finally, in the third part of investigation, the vibration compensation was switched on and a large reduction in the noise on the fringe pattern was obtained, as shown in Figure 8.18. This result shows that the measurement step height was almost identical to the reference measurement. This result illustrates that compensation vibration can be used to overcome environmental disturbance. The cross-section plot of figure 8.18(b) shows the measured step height differs by 0.02 μm from the reference step.

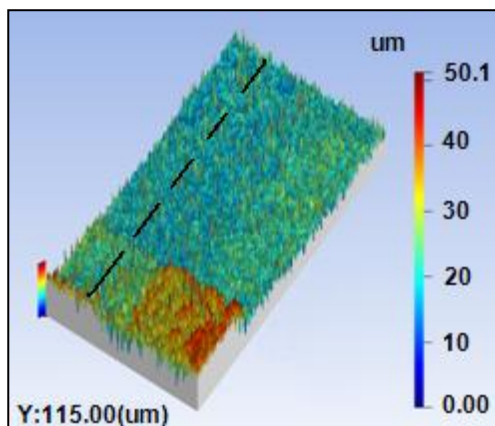


(a) Areal surface view

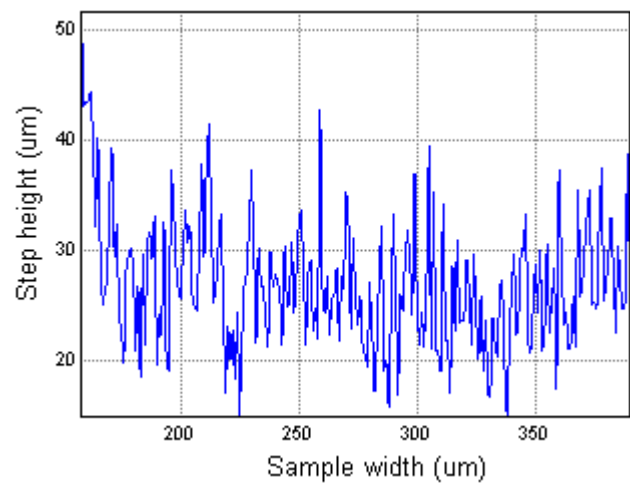


(b) Cross-section view

Figure 8.16 Reference measurement of 5.2 μm step height sample

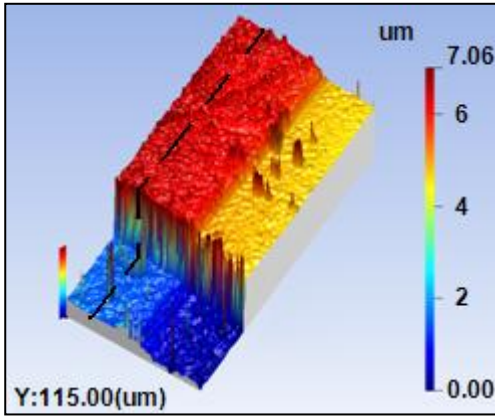


(a) Areal surface view

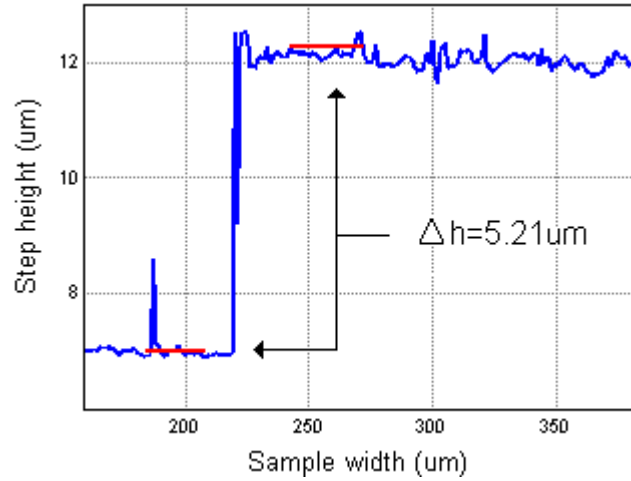


(b) Cross-section view

Figure 8.17 Measurement of 5.2 μm sample with no stabilisation



(a) Areal surface view



(b) Cross-section view

Figure 8.18 Measurement of 5.2 μm step height sample with stabilisation during mechanical disturbance

The magnitude of disturbance attenuation was determined by acquiring the intensity variation of the fringe before and after stabilisation (see figure 8.19). The intensity variation indicated that the fluctuation of the fringe was more than $\lambda/4$. The voltage variation before stabilisation was found to be $V_o = 3\text{V}$, and after stabilisation was $V_{att} = 0.13\text{ V}$, see Figure 8.19. In decibel terms the attenuation was:

$$d\beta = 20\log_{10} \frac{V_{att}}{V_o} \quad (8.26)$$

$$dB = 20\log_{10} \frac{0.13}{3} = -27.3\text{ dB}$$

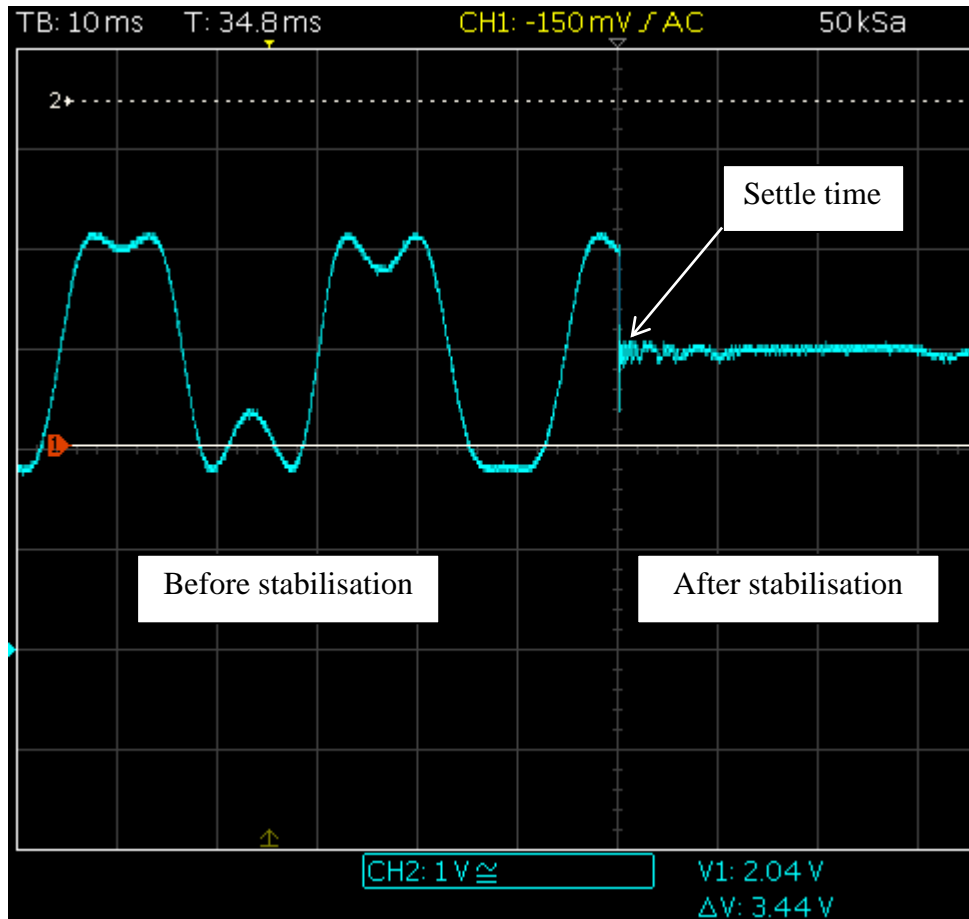


Figure 8.19 Effect of vibration compensation on a 40Hz and 323 nm peak to peak sinusoidal disturbance

To evaluate this attenuation in terms of surface height error, the disturbance vibration needs to be determined first. Since a variation of 2.4 V caused a phase change of approximately π (see Figure 8.19), the vibration magnitude can be determined by re-arranging the form of the interferometer phase equation into,

$$h = \frac{\lambda \phi}{4\pi} \quad (8.27)$$

$$h = \frac{816 * \pi}{4\pi} = 204 \text{ nm}$$

Using linear proportionality approach, the feedback voltage after stabilisation caused surface height variation, h_{err} , equal to 11 nm as illustrated below,

$$\frac{2.4V}{204nm} = \frac{0.13V}{h_{err}} \quad (8.28)$$

This result is in agreement with the measured height difference between the figure 8.16(b) and figure 8.18(b) which is 10 nm.

It was found that the time it took for the reading to stabilise (settle time) was approximately 0.4 ms (see Figure 8.19). This practical settle time is consistent with the theoretical settle time (see Figure 8.13), there being only a 0.1 ms difference. This may be due in part to bandwidth of the electronic circuit and power limitations on the driver supplying the PZT.

The system was tested at range of frequencies from 0 Hz – 350 Hz at the same amplitude of disturbance (i.e. 323 nm peak to peak) (see Figure 8.20). The result shows that the attenuation is decreased by increasing the disturbance frequency. This could be related to the maximum slew rate obtainable in the operation of the servo control loop. Since the PZT is driven directly by LF 353 operational amplifier, which has a slew rate of 8 V/ μ s at ± 15 V, the PZT performance could be limited by the drive electronics.

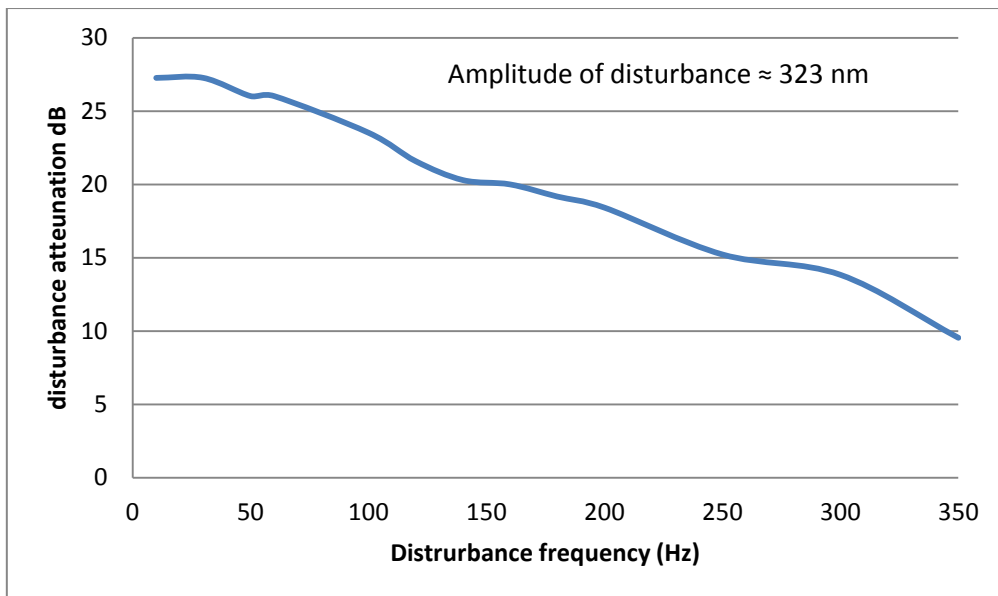


Figure 8.20 Vibration attenuation for a range of disturbance frequencies

8.6 System Repeatability

The repeatability study of WSI system as discussed in Chapter 6 was repeated with the stabilisation on (see Figure 8.21) using the 4.707 μm step sample. The results show a standard deviation of 3.5 nm and an enhancement in the repeatability compared to the non-stabilised repeatability discussed in Chapter 6 which showed a standard deviation of 6.18 nm. Figure 8.21 shows that all the measurements are within the upper and lower limits. Thus the repeatability is within the separation of two standard deviations or 7 nm. This enhancement is mainly due to the stabilisation of the interference pattern obtained from a single pixel. Figures 8.22 and 8.23 illustrate the interference pattern distribution of a single pixel for four sets of measurement before and after stabilisation. No artificial mechanical disturbance has been applied in both cases except the unavoidable environmental noise of the laboratory. The difference in number of peaks may be due to the jumps in fringe order during the locking of the reference mirror on the sample vibration.

Table 8.2 Comparison of repeatability when the stabilisation ON and OFF

Number of measurement	Measured Step height value (μm)		Number of measurement	Measured step height value (μm)	
	Stabilisation OFF	Stabilisation ON		Stabilisation OFF	Stabilisation ON
1	4.72	4.691	8	4.718	4.697
2	4.723	4.688	9	4.718	4.689
3	4.714	4.685	10	4.727	4.692
4	4.717	4.695	11	4.705	4.689
5	4.719	4.693	12	4.71	4.692
6	4.726	4.694	13	4.726	4.697
7	4.72	4.689	14	4.722	4.694

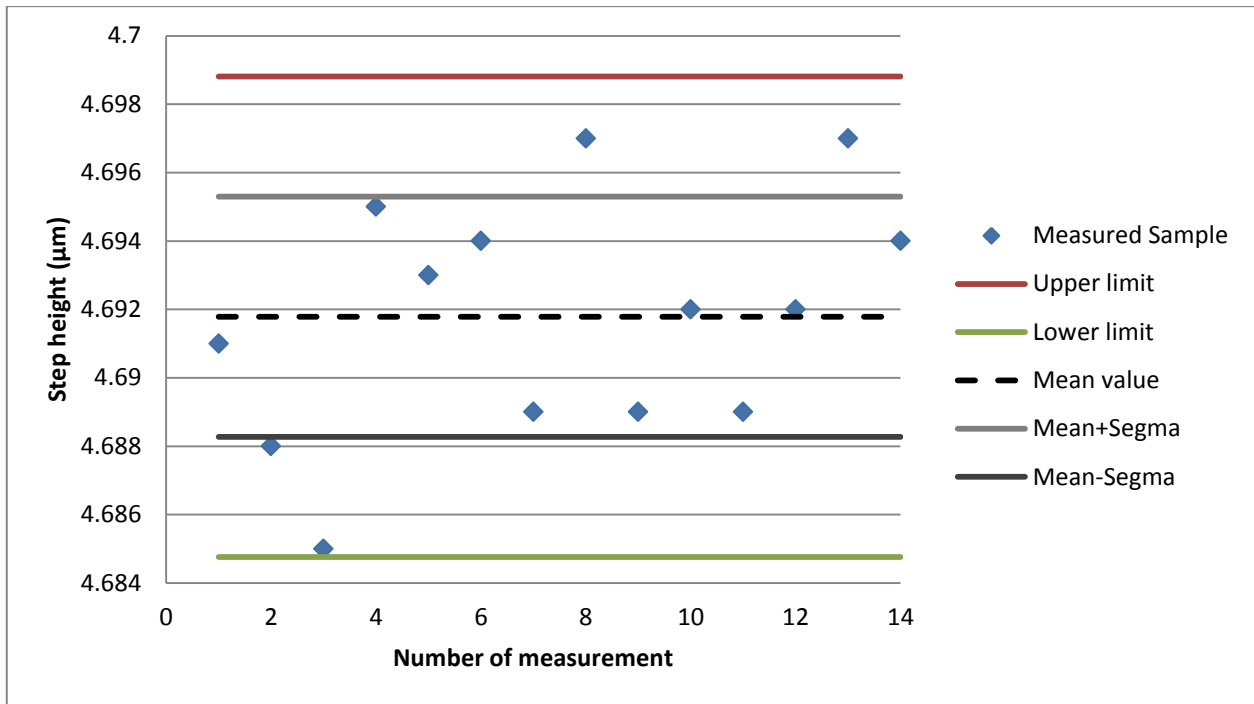


Figure 8.21 The WSI repeatability when the stabilisation is switched on

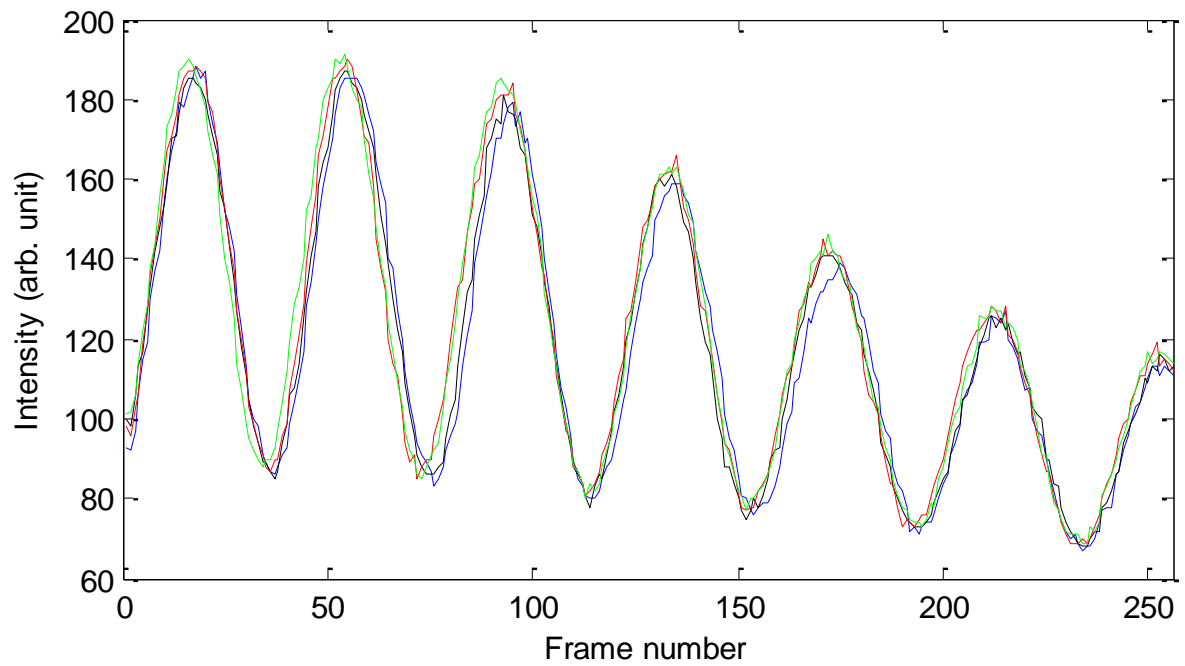


Figure 8.22 The interference pattern of single pixel before stabilisation of background laboratory noise

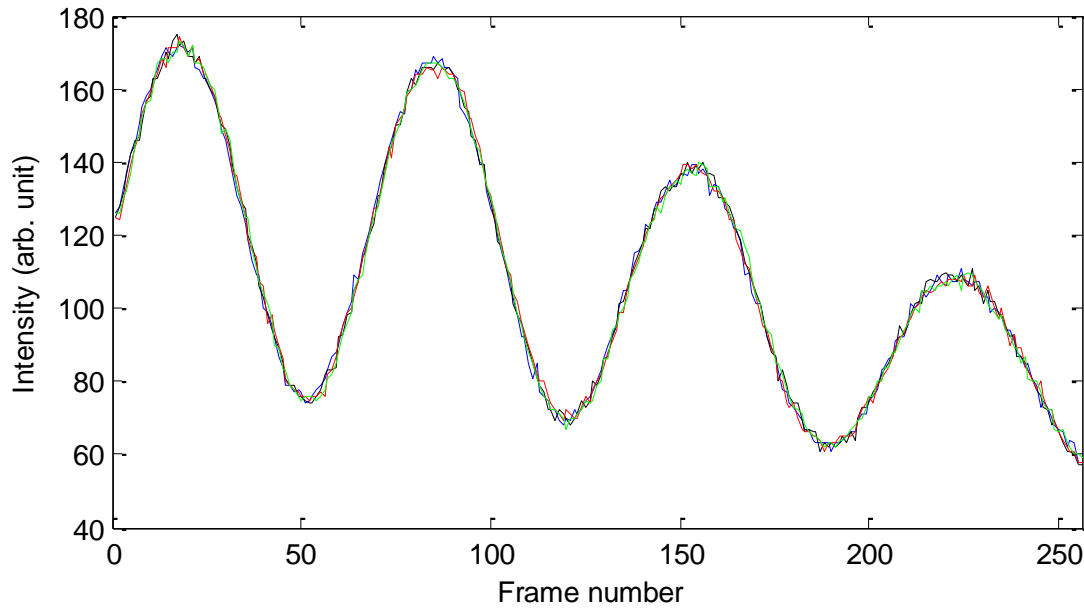


Figure 8.23 The interference pattern of single pixel after stabilisation of background laboratory noise

8.7 Summary

The WSI is stabilised actively by multiplexing a reference interferometer with the WSI such that both interferometers share the same optical path. The purpose of this setup is to stabilise the interferometer using the reference interferometer while the WSI performs the measurement. The compensated WSI should be able to be used for online/in-process measurement on the shop floor.

Stabilisation has been investigated theoretically and practically. Both investigations show that the settle time of the close loop system is no longer than 0.4 msec.

An external mechanical disturbance has been intentionally applied to the system with 323 nm peak to peak amplitude. This disturbance was reduced to approximately 11 nm by the implementation of active stabilisation. The attenuation level obtained using this method is approximately at 27.3 dB for frequencies of up to 40 Hz, after which the attenuation decreases to 9.5 dB at 350 Hz.

9. Discussion, Conclusions and Future Work

9.1 Overview

Integrating measurement and manufacturing processes onto a single platform leads to savings in both time and cost. It was this concept that triggered the development of the interferometry system described in this thesis.

The interferometry was based on using a WSI to measure large discontinuous steps with nanoscale resolution. In order to incorporate this interferometer with the manufacture process, three separate techniques were developed and integrated: (a) Develop fast WSI using AOTF (to provide a fast wavelength scanning over a wide range) and CCD with fast capture rate.; (b) Use GPU technology and CUDA programming to accelerate the computing process and reduce the time of fringe analysis, a technique that greatly improved the overall measurement throughput; (c) Stabilise the interferometer using an active servo controller to cancel out mechanical vibration noise in the workshop environment that adversely affected the interferometer.

It can be seen from the system setup and performance that it is possible to combine two beams having different wavelength regions (visible and IR) and couple them into a Linnik interferometer via a multimode fibre. This technique produced two multiplexed interferometers sharing the same optical path; which meant effective vibration compensation was achievable.

Before starting the measurement, interference fringes are obtained. This can be achieved by aligning the sample within the objective focal depth and permitted surface gradient limit.

To operate the system, the interferometer must first be stabilised. When the fringes are stabilised, the wavelength scanning takes place while simultaneously capturing the frames from using a CCD. After that, the data is sent to a GPU to be analysed using parallel processing. The overall sequence of operation is summarised in Figure 9.1.

The system operation can be controlled by a software platform based on C/C++ program. All the C code and CUDA that presented in this research have been added into this platform using Microsoft foundation classes (MFC). The platform contains toolbars and menus which give the operator the ability to control the WSI and analysis the measured data (see Appendix E).

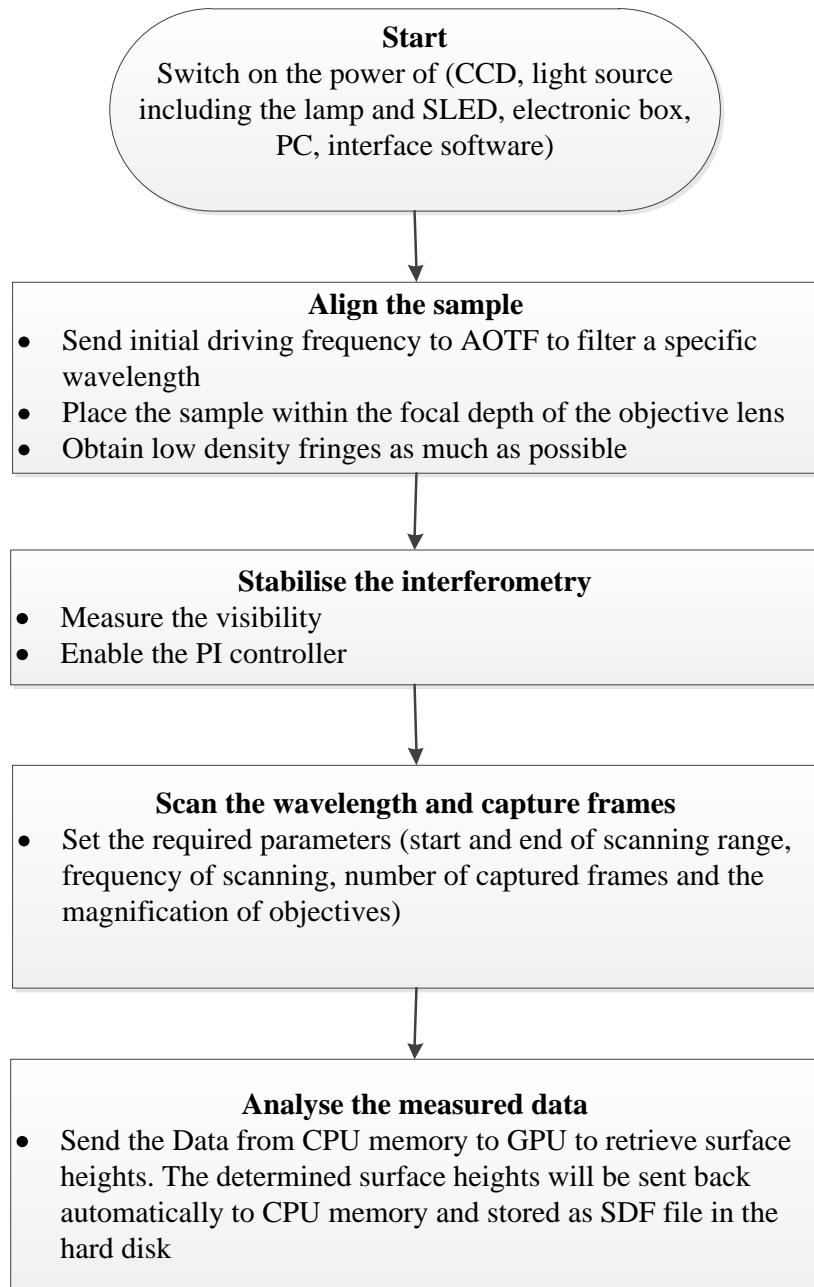


Figure 9.1 Sequence of operation

9.2 System performance

The measurement vertical range is found to be limited by the focal depth of microscope objectives and coherence length of the filtered light. The determined coherence length is approximately $\pm 48\text{ }\mu\text{m}$ while the largest focal depth of 2X objective is $92\text{ }\mu\text{m}$. The vertical resolution depends among other factors on the fringe analysis algorithm. It is found that fringe analysis using a discrete Fourier transform to calculate the phase (algorithm C) has the highest measurement resolution. The typical measurement accuracy obtained for the step height sample was found to be 11 nm .

It can be seen from Chapter 5 that the lateral range was dependent on the microscope objectives and number of pixels in the CCD. The lateral range is found to vary from 0.57 mm (480 pixel width) to 1.88 mm (640 pixel width) at 2X and 5X respectively. The estimated lateral resolution using the Rayleigh criterion was found to be $2.93\text{ }\mu\text{m}$ and $7.35\text{ }\mu\text{m}$ for 2X and 5X respectively.

Stabilising the interferometer during the measurement effectively and efficiently attenuates the noise disturbance. It is found that the attenuation factor was more than 27 dB when the simulated vibration frequency was less than 40 Hz and more than 10 dB when the vibration frequency was increased to around 300 Hz .

Accelerating the fringe analysis has reduced the computing time to less than 1.2 second when 256 frames are considered. As such, the overall measuring time (capture and analysis) is less than 3.8 second .

As discussed in Chapters 6 and 7, the system repeatability and uncertainty are within nanometre values. It is found that the root mean square deviation from the mean of 14 measurements was 6.18 nm before stabilisation. This deviation was reduced to 3.5 nm after the stabilisation was switched on.

The performance specification of WSI instrumentation is listed in Table 9.1

Table 9.1 Specification of WSI

Performance			value	unit
Measurement range	Vertical		96 (maximum)	μm
	Lateral	2X objective	1.882 x 1.412	mm
		5X objective	0.762 x 0.572	mm
Measurement	Vertical		15 nm	nm
Resolution	Lateral	2X objective	7.58	μm
	(estimated)	5X objective	2.98	μm
Measurement time	Capturing time	128 frames	1.3	sec
		256 frames	2.6	sec
	Analysis time	128 frames	0.42	sec
		256 frames	1.15	sec
Stabilisation bandwidth			< 300	Hz
Stabilisation attenuation factor dβ			27 @ low frequency (< 40Hz) 10 @ high frequency (≈ 300 Hz)	
Repeatability of WSI measuring 4.707 μm standard step sample at approximately 15 μm height from the virtual reference plane.		Stabilisation on	7	nm
		Stabilisation off	12.36	nm

9.3 Major Contribution

The major contribution to knowledge in this thesis can be summarised as the following:

- The implementation and operation of a fast WSI instrument which can be used for surface texture measurement. The operation of this instrument is based on scanning the wavelength of a broadband tungsten-halogen lamp from 683.42 to 590.98 nm using an AOTF in order to shift the phase of the interference fringes in a linear manner with no mechanically movement parts.
- Verification of the performance of a WSI by resolving step height and V-groove surface structured samples. This thesis shows that it is possible to measure V-groove samples with large surface gradient by simply rotating and tilting the instrument to face the groove side wall since there is no any mechanical movement preventing to perform this action.
- A comparison study of four algorithms used to analyse the interference fringe pattern obtained from WSI and evaluate their accuracy. This study gives a mathematical description of the algorithm resolution which is a function of the optical path difference, wavelength scanning range and wavelength scanning step.
- Establishment of a parallel CUDA programming structure to accelerate the computing process of fringe analysis by generating a massive numbers of threads, equal to the number of the CCD pixels, and executing the program code using the many cores of a GPU. For instance, the computing time is reduced from 32 s to 1.2 s for 256 frames of 640 x 480 pixels.
- Development of an opto-electronic feedback control system to eliminate environmental noise with frequencies up to 300 Hz. This thesis also introduces a mathematical model to describe the close-loop control system theoretically.
- Demonstration of the WSI for embedded surface metrology to resolve large discontinuous step height sample of tens of micrometres in a simulated manufacturing environment.

In addition, this work has been published in four journal papers and five conferences. This instrument is also covered by an international patent (WO/2010/082066), as stated in the ‘Publications and Awards’ section.

9.4 Conclusions

WSI was investigated with a view to producing an instrument that is fast enough for real time use that can measure large discontinuous step heights and robust enough to be used in the manufacturing processes. The aim was to deliver non-contact measurement over a large micro-scale range with nano-scale resolution for in-process measurement applications. The research carried out on developing the WSI aimed to improve the current limited state of the art in embedded metrology. This thesis proposes a surface measurement technique based on wavelength scanning interferometry (WSI) with active control of environmental vibration and accelerated fringe analysis. Surface measurement accurate to nanometre scales have been achieved for sample step heights in the micrometre range. There is no need for optical path (mechanical) scanning as found in conventional white-light vertical scanning interferometry. It was found that an AOTF can provide a linear phase shift over a wide wavelength scanning range. The spectral linewidth of the filtered wavelength using AOTF with large acceptance angle can provide a coherence length of nearly 100 μm . A near IR SLED sharing the same optical path of the WSI allows the introduction of a vibration compensation feedback loop. A PZT is used to convert the electric control signal to mechanical movement in order to compensate for the error in the OPD due to environmental vibration disturbance. The experiment shows that the noise attenuation factor is more than 27 dB at less than 40 Hz vibration frequency.

The measurement speed is restricted only by the frame rate of the CCD camera and the data processing speed. A 100 frames/second capture rate was employed in the WSI to achieve fast

acquisition of the required data. Typically 256 frames are needed to achieve accuracy with nanometre values.

The net measurement throughput was further improved by reducing the computation time required for the fringe analysis. It was found that using a commercial GPU with 240 cores can accelerate the computing process by up to 27.6 times compared to multi-core CPU computing. An areal surface, consisting of 640x480 pixels, was analysed in less than 1.2 second. It was also found also that single floating point operations of the GPU satisfy the accuracy requirement of the WSI.

As demonstrated in Chapter 6, the accuracy of WSI measurement can be improved by considering a suitable algorithm to calculate the phase shift more rapidly and more precisely. It is found that the height resolution was different for each algorithm. The computing resolution in algorithm A, which determines the phase shift from the interference pattern frequency using a simple FFT, is limited by $\lambda_s/2$. This resolution is improved by a factor $(F + \varepsilon)/N$ in algorithm C, which determines the phase by analysing the interference pattern using an FFT and inverse FFT. According to this, the measurement accuracy is improved from the micron to the nanoscale.

The measurement resolution depends on the OPD as well as the wavelength scanning range, $\Delta\lambda$ and scanning step, $\delta\lambda$. It was found that the resolution is increased with surfaces that are close to the point of zero optical path difference. The S_{5z} value within the surface profile is reduced by increasing the measurement resolution. This enhancement in the measurement resolution is due in part to increasing the sampling rate. However, if the measurement algorithm has a computing resolution below that of the surface S_{5z} , then no surface roughness information can be observed at any sampling rate.

9.5 Future Work

Further work needs to be carried out to optimise the WSI performance in terms of accuracy, range of vertical measurement, speed of measurement and robustness as described in the following subsections.

9.5.1 Improvement of accuracy and measurement range

The AOTF that was used throughout this thesis has large aperture (12 x 12) mm that yields an approximately 2 nm linewidth over the suggested scanning range. This AOTF can be replaced by one with a reduced aperture to achieve sub-nanometre linewidth. This narrow linewidth can improve the temporal coherence and measurement accuracy. An example of an AOTF with a small aperture is produced by Gooch and Housego (TF560-280-1-5-NT2). This AOTF can provide an approximately 0.3 nm linewidth over the visible wavelength region. This narrow linewidth can provide temporal coherence greater than 500 μm . Also the tunable wavelength step, $\delta\lambda$ can be reduced to smaller values to improve the measurement resolution as explained in chapter 6. The main challenge however for this kind of AOTF is to obtain a collimated broadband light to satisfy the acceptance angle of the AOTF aperture. The acceptance angle for the given example is approximately 2.5 mrad. This limitation makes the use of halogen-tungsten lamp inappropriate for this system. A super-continuum light source can replace the lamp to produce a powerful wavelength scanning light source. This light source can also provide sufficient light intensity throughput to make it suitable for use with a high capture rate video camera.

9.5.2 Accelerate the capturing time

The CMOS camera can be used to reduce the capture time to sub-second values. A capture rate of more than 500 frames/second can be achieved from commercial CMOS cameras. This would reduce the capture time to less than half a second.

9.5.3 Accelerate the computing time

It can be seen from Chapter 7 that the acceleration factor is decreased as the number of captured frames is increased. This can be overcome by using a GPU supported by a larger number of cores and larger memory capacity such as GTX690 from NVIDIA which has 3072 cores and 4 GB. This could provide a high acceleration factor for a higher number of captured frames.

9.5.4 Increase the stabilisation bandwidth

As discussed in Chapter 8, the noise attenuation factor decreases at higher frequencies. This may be due to the electric power limitation of the IC used in the PI circuitry. An investigation can be carried out using a high power PZT amplifier to find the effectiveness of stabilisation at high vibration frequencies and amplitudes. An example of such an amplifier is E-617 from Physik Instrumente. This PZT amplifier can provide a peak power of up to 280 W and 2A of peak current. The bandwidth of this amplifier is approximately 3.5 kHz which is much higher than the environmental vibration found in the workshop. Such an amplifier could increase the performance of the stabilisation scheme to a maximum (i.e. 15 μm).

10. References

- ABBOTT, E. J. & FIRESTONE, F. A. 1933. Specifying Surface Quality: A method based on Accurate measruement and comparsion *Mechanical Engineering*, 55, 569-573.
- ADHIKARI, R. 2004. *Sensitivity and Noise Analysis of 4 Km Laser Interferometric Gravitational Wave Antennae*. Ph.D, Massachuestts Institute of Technology.
- AEROTECH, I. 2010a. *AGR Mechanical-Bearing Gear-Drive Rotary Stage* [Online]. Available: <http://www.aerotech.com/product-catalog/stages/rotary-stages/agr.aspx> [Accessed Oct 2010].
- AEROTECH, I. 2010b. *PRO165 Mechanical-Bearing Ball-Screw Linear Stage* [Online]. Available: <http://www.aerotech.com/product-catalog/stages/linear-stages/pro165.aspx> [Accessed Oct 2010].
- ALVAREZ, L. & SIQUEIROS, A. J. M. 2010. Scanning Probe Microscopy *Formatex*, 1302-1309.
- ANAND LAL SHIMPI, D. W. 2008. *NVIDIA's 1.4 Billion Transistor GPU: GT200 Arrives as the GeForce GTX 280 & 260* [Online]. Available: <http://www.anandtech.com/show/2549/5> [Accessed April 2012].
- ANDOR-TECHNOLOGY. 2012. *CCD Frame Rates* [Online]. Andor Technology Ltd. Available: http://www.andor.com/learning/digital_cameras/?docid=323 [Accessed].
- APC INTERNATIONAL. 2011. *Piezoelectric Constants* [Online]. APC International, Ltd. Available: <http://www.americanpiezo.com/knowledge-center/piezo-theory/piezoelectric-constants.html> [Accessed July 2011].
- BADAMI, V. G., LIESENER, J., EVANS, C. J. & GROOT, P. D. 2012. *Evaluation of the measurement performance of a coherence scanning microscope using roughness specimens* [Online]. Available: http://www.zygo.com/library/papers/aspe_111115.pdf [Accessed Sept 2012].
- BALSUBRAMANIAN, N. 1980. Optical system for surface topography measurement. *U.S. patent 4,340,306*.
- BENNETT, J. M. & DANCY, J. H. 1981. Stylus profiling instrument for measuring statistical properties of smooth optical surfaces. *Applied Optics*, 20, 1785-1802.
- BENNETT, J. M. & MATTSSON, A. L. 1989. *Introduction to Surface Roughness and Scattering*, Washington, D.C.
- BHUSHAN, B. 1999. *Principles and Applications of Tribology*, John Wiley & Sons, Inc.
- BHUSHAN, B. 2001. *Modern Tribology Handbook*, CRC Press LLC.
- BINNIG, G. & ROHRER, A. H. 1983. Scanning Tunneling Microscopy. *North-Holland (Surface Science)*, 126 236-244.
- BLUNT, L. & JIANG, X. 2003. *Advanced Techniques for Assessment Surface Topography*, Kogan Page Science.
- BUICE, E. S. 2007. *Implementation of A Dynamuc Positioning Machine For Nano-scale Engineering*. PhD, The University of North Carolina.
- CABRAL, A. & REBORDÃO, J. 2007. Accuracy of frequency-sweeping interferometry for absolute distance metrology. *Optical Engineering* 46.
- CARRÉ, P. 1966. Installation et utilisation du comparateur photoelectrique et interferentiel du Bureau International des Poids et Mesures. *Metrologia*, 2, 13-23.
- CHANG, L.-W., CHIEN, P.-Y. & LEE, C.-T. 1999. Measurement of absolute displacement by a double-modulation technique based on a Michelson interferometer. *APPLIED OPTICS* 38, 2843-2847.
- CHARETTE, P. G. & HUNTER, I. W. 1996. Robust phase-unwrapping method for phase images with high noise content. *Applied Optics*, 35, 3506-3513.

- CHEN, C. J. 2008. *Introduction to Scanning Tunneling Microscopy*, OXFORD UNIVERSITY PRESS.
- CHURCH, E. L. & ZAVADA, J. M. 1975. Residual surface roughness of diamond-turned optics. *Applied Optics*, 14.
- CONCHELLO, J.-A., KIM, J. J. & HANSEN, A. E. W. 1994. Enhanced three-dimensional reconstruction from confocal scanning microscope images. *Applied Optics*, 33, 3740-3750.
- CUI, F. K., SONG, Z. B., WANG, X. Q., ZHANG, F. S. & LI, Y. 2010. Study on laser Triangulation Measurement Principle of Three Dimensional Surface Roughness *Advanced Materials Research*, 136, 91-94.
- DAVILA, A., HUNTLEY, J. M., PALLIKARAKIS, C., RUIZ, P. D. & COUPLAND, J. M. 2012. Simultaneous wavenumber measurement and coherence detection using temporal phase unwrapping. *Applied Optics*, 51, 558-567.
- DEJIAO, L., JIANG, X., XIE, F., ZHANG, W., ZHANG, L. & BENNION, I. 2004. High stability multiplexed fibre interferometer and its application on absolute displacement measurement and on-line surface metrology *Optics Express*, 12, 5729-5734.
- DORSCH, R. G., HAUSLER, G. & HERRMANN, A. J. M. 1994. Laser triangulation: fundamental uncertainty in distance measurement. *Applied Optics*, 33.
- ELSON, J. M. & BENNETT, J. M. 1979. Relation between the angular dependence of scattering and the statistical properties of optical surfaces. *Optical Society of America*, 69, 31-47.
- EWING, G. BUCHER, J. W. C. 1999. Characterisation of an Acousto-optic Tunable Filter and Use in Visible Spectrophotometry. *Applied Spectroscopy*, 53, 603-611.
- GAO, F., LEACH, R., PETZING, J. & COUPLAND, J. M. 2008. Surface Measurement Errors using Commercial Scanning White Light Interferometers *Meas. Sci. Technol.*, 19.
- GIURGIUTIU, V. & LYSHEVSKI, S. 2004. *Micromechatronics: Modeling, Analysis, and Design with Matlab*, CRC Press.
- GROOT, P. D., LEGA, X. C. D., KRAMER, J. & TURZHITSKY, M. 2002. Determination of fringe order in white-light interference microscopy. *Applied Optics*, 41, 4571-4578.
- GROOVER, M. P. 2007. *Fundamentals of Modern Manufacturing: Materials, Processes, and Systems*, John Wiley & Sons, Inc.
- HARASAKI, A., SCHMIT, J. & WYANT, J. C. 2001. Offset of coherent envelope position due to phase change on reflection. *Applied Optics*, 40, 2102-2106.
- HARIHARAN, P. 1991. Optical interferometry. *Reports on Progress in Physics*, 54, 352.
- HARIHARAN, P., OREB, B. F. & EIJU, T. 1987. Digital phase-shifting interferometry: a simple error-compensating phase calculation algorithm. *Applied Optics*, 26, 2504-2506.
- HAYES, J. 2002. *Dynamic interferometry handles vibration* [Online]. Laser Focus World. Available: <http://www.laserfocusworld.com/articles/print/volume-38/issue-3/features/vibration-control/dynamic-interferometry-handles-vibration.html> [Accessed].
- HECHT, E. 2002. *Optics*, San Francisco, Addison Wesley.
- HOUSEGO, G. A. 2010. Specifications: TF625-350-2-12-BR1. In: LTD, G. H. (ed.).
- HOUSEGO, G. A. 2011. Specifications: Model number 64020-200-2ADSDFS-A. In: LTD, G. H. (ed.) *Document number: 56A18640A*.
- HUNTLEY, J. M. 1989. Noise-immune phase unwrapping algorithm *Applied Optics*, 28, 3268-3270
- ISHII, Y. 1999. Wavelength-Tunable Laser-Diode Interferometer *OPTICAL REVIEW* 6, 273-283
- ITOH, K. 1982. Analysis of the phase unwrapping algorithm *Applied Optics*, 21, 2470
- JIANG, F., LI, J., YAN, L., SUN, J. & ZHANG, S. 2010. Optimizing end-milling parameters for surface roughness under different cooling/lubricating conditions. *Int J Adv Manuf Technol* 51, 841-851.

- JIANG, X., SCOTT, P., WHITEHOUSE, D. & AND BLUNT, L. 2007a. Paradigm shifts in surface metrology. Part I. Historical philosophy. *Proceedings of the Royal Society A*. 463, 2049-2070.
- JIANG, X., SCOTT, P., WHITEHOUSE, D. & AND BLUNT, L. 2007b. Paradigm shifts in surface metrology. Part II. The current shift. *Proceedings of the Royal Society A*. 463, 2071-2099.
- JIANG, X. & WHITEHOUSE, D. J. 2012. Technological shifts in surface metrology. *CIRP Annals - Manufacturing Technology*, <http://dx.doi.org/10.1016/j.cirp.2012.05.009>.
- KAFRI, O. 1989. Fundamental limit on accuracy in interferometry. *Optics Letters*, 14, 657-658.
- KANG, C. M., WOO, H. G., CHO, H. S., HAHN, J. W. & LEE, J. Y. 2001. Inspection system for microelectronics BGA package using wavelength scanning interferometry. *proc. SPIE*, 4564.
- KANG, M.-G., LEE, S.-Y. & KIN, S.-W. 1999. Self-Compensation of PZT Errors in White Light Scanning Interferometry. *Optical Society of Korea*, 3, 35-40.
- KAPLONEK, W. & LUKIANOWICZ, C. 2012. *Chapter 1: Coherence Correlation Interferometry in Surface Topography Measurements* [Online]. Available: <http://www.intechopen.com/books/recent-interferometry-applications-in-topography-and-astronomy> [Accessed].
- KATO, J.-I. & YAMAGUCHI, L. 2000. Phase-Shifting Fringe Analysis for Laser Diode Wavelength-Scanning Interferometer *OPTICAL REVIEW*, 7, 158-163.
- KEMAO, Q., FANGJUN, S. & XIAOPING, W. 2000. Determination of the best phase step of the Carré algorithm in phase shifting interferometry. *Meas. Sci. Technol.*, 11, 1220-1223.
- KIKUTA, H., IWATA, K. & NAGATA, R. 1986. Distance measurement by the wavelength shift of laser diode light. *Applied Optics*, 25, 2976-2980.
- KIRK, D. B. & HWU, W.-M. W. 2010. *Programming Massively Parallel Processors*, Elsevier.
- KUWAMURA, S. & YAMAGUCHI, I. 1997. Wavelength scanning profilometry for real-time surface shape measurement. *Applied Optics*, 36, 4473-4482.
- LARKIN, K. G. 1996. Efficient nonlinear algorithm for envelope detection in white light interferometry. *J. Opt. Soc. Am. A*, 13, 832-843.
- LEACH, R. 2010. *Fundamental principles of engineering nanometrology*, Amsterdam, William Andrew.
- LEACH, R. K. 2000. *NanoSurf IV: Traceable measurement of surface texture at the nanometre level*. Ph.D, The university of Warwick.
- LIN, W.-W., SHIH, S.-T., CHEN, M.-H. & HUANG, S.-C. 1992. The Transfer Function of the Thin Shell PZT Ceramic Cylinder as a Phase Modulator in Fiber Optics Sensors. *SPIE*, 1814, 130-141.
- LIU, J.-M. 2005. *Photonic Devices*, Cambridge University Press.
- LU, S.-H., CHIUEH, C.-I. & LEE, C.-C. 2002. Differential wavelength-scanning heterodyne interferometer for measuring large step height. *Applied Optics*, 41, 5866-5871.
- LUO, X., CHENG, K. & WEBB, D. 2005. *Design of ultraprecision machine tools with applications to manufacture of miniature and micro components* [Online]. Available: <http://bura.brunel.ac.uk/bitstream/2438/1352/5/Paper-Cheng.doc.txt> [Accessed Aug 2012].
- LYONS, K. W. 2007. Integration, interoperability, and information management: What are the key issues for nanomanufacturing? . *Proc. SPIE*, 6648.
- MABOUDIAN, R. 1997. Surface process in MEMS technology. *Surface Science Reports*, 30, 207-267.
- MATSUMOTO, H. 1986. Synthetic interferometric distance-measuring system using a CO₂ laser. *APPLIED OPTICS*, 25, 493-498.

- MCCLANAHAN, C. 2010. *History and Evolution of GPU Architecture* [Online]. Available: <http://mcclanahoochie.com/blog/2011/03/the-history-and-evolution-of-gpu-hardware/> [Accessed Nov 2011].
- MILLERD, J. E., BROCK, N. J., HAYES, J. B. & WYANT, J. C. 2004. Instantaneous phase-shift, point-diffraction interferometer. *Proceedings of SPIE*, 5531, 264-272.
- MITUTOYO. 2010. *Microscope units and objectives* [Online]. Available: http://www.mitutoyo.com/pdf/E4191-378_010611.pdf [Accessed Sept 2009].
- MTI INSTRUMENTS INC. 2012. *Laser Triangulation Sensors* [Online]. Available: <http://www.mtiinstruments.com/products/lasertriangulation.aspx> [Accessed July 2012].
- NGOI, B. K. A., VENKATAKRISHNAN, K. & SIVAKUMAR, N. R. 2001. Phase-shifting interferometry immune to vibration. *Applied Optics*, 40, 3211-3214.
- NIST. 2002. *Acceptable Data Format* [Online]. Available: <http://pml.nist.gov/VSC/jsp/DataFormat.jsp> [Accessed July 2012].
- NVIDIA¹. 2012. *Specifications: GTX285* [Online]. Available: http://www.nvidia.co.uk/object/product_geforce_gtx_285_uk.html [Accessed Dec 2010].
- NVIDIA². 2012. *History of GPU Computing*. Available: http://www.nvidia.com/object/cuda_home_new.html.
- NVIDIA. 2007. *CUDA CUFFT Library* [Online]. Santa Clara: NVIDIA Corporation. Available: http://moss.csc.ncsu.edu/~mueller/cluster/nvidia/0.8/NVIDIA_CUFFT_Library_0.8.pdf [Accessed 20 May 2011].
- NVIDIA. 2008. *NVIDIA CUDA Compute Unified Device Architecture: programming guide 2.0* [Online]. Available: http://developer.download.nvidia.com/compute/cuda/2_0/docs/NVIDIA_CUDA_Programming_Guide_2.0.pdf [Accessed].
- NVIDIA 2011. *NVIDIA CUDA C: Programming Guide*. NVIDIA Corporation.
- NVIDIA. 2012a. *Specifications: GTX285* [Online]. Available: http://www.nvidia.co.uk/object/product_geforce_gtx_285_uk.html [Accessed Dec 2010].
- NVIDIA. 2012b. *History of GPU Computing* [Online]. Available: <http://www.nvidia.com/object/what-is-gpu-computing.html> [Accessed 2012 Jan].
- OGATA, K. 2002. *Modern Control Engineering*, Prentice-Hall, Inc.
- OHBA, R., UEHIRA, L. & KAKUMA, S.-I. 1990. Interferometric determination of a static optical path difference using a frequency swept laser diode. *Meas. Sci. Technol.*, 1, 500-504.
- OWENS, J. D., HOUSTON, M., LUEBKE, D., GREEN, S., STONE, J. E. & PHILLIPS, J. C. 2008. GPU Computing. *Proceedings of the IEEE*, 96, 879-899.
- PERSSON, U. 1998. In-process measurement of surface roughness using light scattering. *Elsevier Science S.A, Wear* 215, 54-58
- PETZING, J., COUPLAND, J. & LEACH, R. 2010. *Good Practice Guide No. 116: The Measurement of Rough Surface Topography using Coherence Scanning Interferometry*, NPL.
- PHYSIK. 2009. *Piezoelectrics in Positioning* [Online]. **Physik Instrumente (PI) GmbH & Co.** Available: http://www.physikinstrumente.com/en/pdf_extra/2009_PI_Piezo_University_Designing_with_Piezo_Actuators_Tutorial.pdf [Accessed April 2012].
- PI-CERAMIC. 2010a. *P840_Datasheet* [Online]. Available: http://www.physikinstrumente.com/en/pdf/P840_Datasheet.pdf [Accessed Novembre 2010].
- PI-CERAMIC. 2010b. *Typical Parameters of Piezoelectric Ceramics* [Online]. PI Ceramic GmbH. Available: http://www.piceramic.com/piezo_materials_2.php [Accessed July 2011].
- PI CERAMIC. 2010. *P840_Datasheet* [Online]. Available: http://www.physikinstrumente.com/en/pdf/P840_Datasheet.pdf [Accessed Novembre 2010].

- RIEMER, O. 2011. *Advances in Ultra Precision Manufacturing* [Online]. Available: <http://www.jspe.or.jp/english/sympo/2011s/2011s-1-7.pdf> [Accessed 2012 July].
- SALEH, B. E. A. & TEICH, M. C. 2007. *Fundamentals of Photonics*, Wiley-Interscience.
- SANDOZ, P., DEVILLERS, R. & PLATA, A. 1997. Unambiguous profilometry by fringe order identification in white-light phase-shifting interferometry. *J. Mod. Opt.*, 44, 519-534.
- SCHREIBER, H. & BRUNING, J. H. 2007. *Chapter 14: Optical Shop Testing*, John Wiley & Sons, Inc.
- SCHWIDER, J., BUROW, R., ELSSNER, K., GRZANNA, J., R., S. & MERKEL, K. 1983. Digital wave-front measuring interferometry: some systematic error sources. *Applied Optics*, 22, 3421-3432.
- SCHWIDER, J. & ZHOU, L. 1994. Dispersive interferometric profilometer. *Optics Letters*, 19, 995.
- SEMWOGERERE, D. & WEEKS, E. R. 2005. *Confocal Microscopy*, Taylor & Francis.
- SHI, K., LI, P., YIN, S. & LIU, Z. 2004. Chromatic confocal microscopy using supercontinuum light *Optics Express*, 12, 2096- 2101.
- SMYTHE, R. & MOORE, R. 1984. Instantaneous phase measuring interferometry. *Optical Engineering*, 23, 361-365.
- SNYDER, J. J. 1980. Algorithm for fast digital analysis of interference fringes. *Applied Optics*, 19, 1223-1225.
- SPRING, K. R., FELLERS, T. J. & DAVIDSON, M. W. 2009. *Resolution and Contrast in Confocal Microscopy* [Online]. Available: <http://www.olympusconfocal.com/theory/resolutionintro.html> [Accessed May 2012].
- STANFORD, J. & HAGAN, P. 2009. An Assessment of the Impact of Stylus Metallurgy on Cerchar Abrasiveness Index. *Coal Operators' Conference*, 348-355.
- STONE, J. A., STEJSKAL, A. & HOWARD, L. 1999. Absolute interferometry with a 670-nm external cavity diode laser. *APPLIED OPTICS* 38.
- SVELTO, O. 1998. *Principles of Lasers*, New York, Plenum Press.
- TAKEDA, M., HIDEKI, I. & KOBAYASHI, S. 1982. Fourier-transform method of fringe-pattern analysis for computer-based topography and interferometry. *OSA Optical Society of America*, 72, 156-160.
- TAKEDA, M. & YAMAMOTO, H. 1994. Fourier-transform speckle profilometry: three-dimensional shape measurements of diffuse objects with large height steps and/or spatially isolated surfaces. *Applied Optics*, 33 7829-7837.
- TAYLOR-HOBSON 2005. Talysurf CCI 6000.
- THIEL, J., PFEIFER, T. & HARTMANN, M. 1995. Interferometric measurement of absolute distances of up to 40 m. *Measurement*, 16, 1-6.
- THIRUNAVUKKARASU 2007. Optical Sensing Limits in Contact and Bending Mode Atomic Force Microscopy. *Experimental Mechanics* 47, 841-844.
- THOMAS, T. R. 1999. *Rough Surfaces*, London.
- TIZIANI, H. J. & UHDE, A. H.-M. 1994. Three-dimensional image sensing by chromatic confocal microscopy. *Applied Optics*, 33, 1838-1843.
- UCHINO, K. & GINIEWICZ, J. 2003. *Micromechatronics*, Marcel Dekker, Inc. .
- UPS2. 2011. *Pyramidal structures* [Online]. Available: <http://www.ups2.co.uk/default.asp?chapterid=59&langid=1> [Accessed Jun 2011].
- VACHARANUKUL, K. & MEKID, S. 2005. In-process dimensional inspection sensors. *Measurement* 38, 204-218.
- VEECO, I. 2012. *Wyko NT Series Optical Profilers: NT9300* [Online]. Available: [http://www.veeco.com/pdfs/brochures/B506-Rev_A7-Wyko_NT_Series_Optical_Profilers-Brochure_\(HiRes\).pdf](http://www.veeco.com/pdfs/brochures/B506-Rev_A7-Wyko_NT_Series_Optical_Profilers-Brochure_(HiRes).pdf) [Accessed July 2012].

- VORBURGER, T. V., MARX, E. & LETTIERI, T. R. 1993. Regimes of surface roughness measurable with light scattering. *Applied Optics*, 32.
- WALSH, C. J. 1987. Measurements of absolute distances to 25 m by multiwavelength CO₂ laser interferometry. *APPLIED OPTICS*, 26, 1680-1687.
- WARD, J., PANNELL, C. N., WACHMAN, E. S. & SEALE, W. 2012. Applications of acousto-optic devices for spectral imaging systems Available: <http://www.neostech.com/PDFs/aotf.pdf>.
- WHITEHOUSE, D. J. 2011. *Handbook of Surface and Nanometrology*, Taylor and Francis Group.
- WHITEPAPER, F. 2009. NVIDIA's Next Generation CUDA Compute Architecture.
- WYANT, J., KOLIOPOULOS, C., BHUSHAN, B. & GEORGE, O. 1984. An Optical Profilometer for Surface Characterization of Magnetic Media. *ASLE Transactions*, 27, 101-113.
- WYANT, J. C. 2000. Improved vertical-scanning interferometry. *Applied Optics*, 39, 2107-2115.
- WYANT, J. C. 2002. White Light Interferometry. *Proceedings of SPIE*, 4737, 98-107.
- XIAOLI, D. & KATUO, S. 1998. High-accuracy absolute distance measurement by means of wavelength scanning heterodyne interferometry. *Measurement Science and Technology*, 9, 1031.
- XU, J. & STROUD, R. 1992. *Acousto-Optic Devices: Principles, Design, and Applications*, John Wiley & Sons, Inc.
- YAMAGUCHI, I., YAMAMOTO, A. & KUWAMURA, S. 1998. Speckle decorrelation in surface profilometry by wavelength scanning interferometry. *Applied Optics*, 37, 6721-6828.
- YAMAMOTO, A., KUO, C.-C., SUNOUCHI, K., WADA, S., YAMAGUCHI, I. & TASHIRO, H. 2001. Surface Shape Measurement by Wavelength Scanning Interferometry Using an Electronically Tuned Ti:Sapphire Laser. *OPTICAL REVIEW* 8, 56-63.
- YANG, H.-J., NYBERG, S. & RILES, K. 2007. High-precision absolute distance measurement using dual-laser frequency scanned interferometry under realistic conditions. *Nuclear Inst. and Methods in Physics Research, A*, 575, 395-401.
- YOUNG, R. D., VORBURGER, T. V. & TEAGUE, E. C. 1980. In-Process and On-Line Measurement of Surface Finish. *Annual of the CIRP*, 29.
- ZYGO. 2012. *NewView™ 700s* [Online]. Available: <http://www.zygo.com/?/met/profilers/newview700/> [Accessed Aug 2012].

11. Appendix A

A.1 AOTF Driving Process

The AOTF driver (Gooch and Housego, 64040-200-2AMDFS-A) is a digitally controlled frequency synthesizer. The frequency is selected by TTL logic 30 bit binary word. The binary word is input through a 37 pin D sub connector. The remaining 7 pins are divided into 3 control lines and 4 not connected lines. The three control lines consist of latch, digital modulation and reset. The Latch keeps the current output frequency from changing while next input word is generated. The latch signal activates the new word, when it goes low. Both the digital modulation and reset lines can be used to disable the output of the driver; however the reset is used also to clear the stored frequency information. Figure A.1 shows that timing diagram concept of the driver.

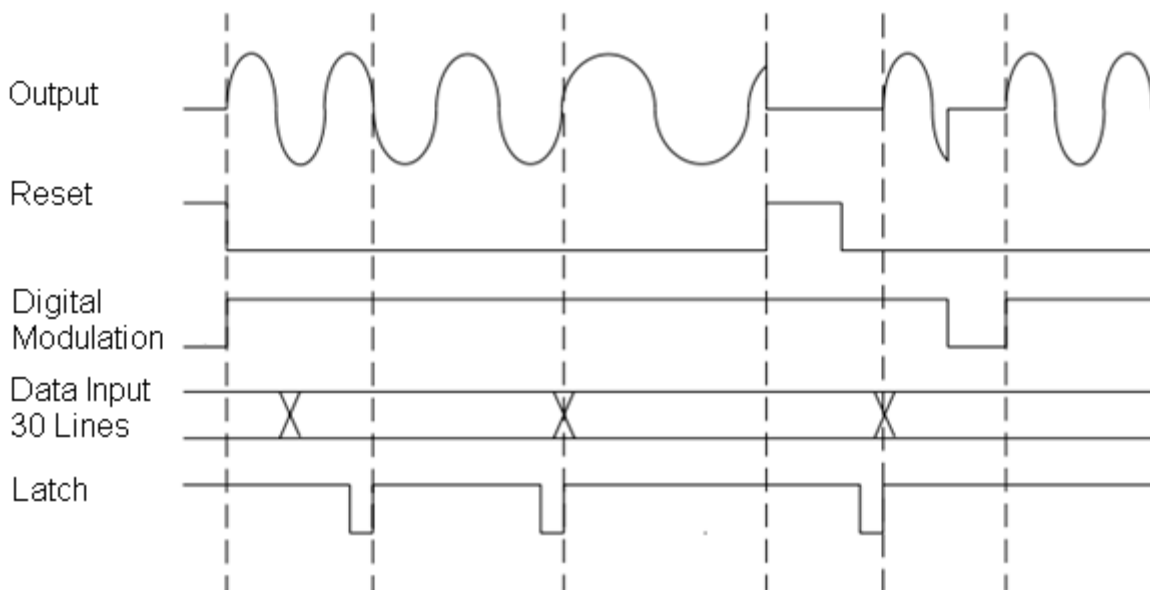


Figure A.1 Timing and waveform diagram for frequency synthesiser

The binary word can select frequency range from Hz scale to MHz scale. Since the required driving frequency for this research is from 80.07 MHz to 95.70 MHz with 0.061 MHz incremental step, the first 15 pins are grounded and the binary increment is started from pins 16.

Most commercial data acquisition cards have a limited number of digital output channels which make it inconvenient to send 15 binary bits in a parallel manner, thus a binary counter electronic circuit has been implemented to count generated digital pulses and convert them to binary bits with TTL voltage (see figure A.2). The generated train of pulses is sent by National Instruments data acquisition card NI-DAQ using the supplied NI-DAQmx C/C++ functions.

The serial to parallel converter circuit consists of two ripple-carry binary counters and one inverter logic gate. The binary counters are used to produce the parallel word while the inverter logic gate is used to produce the latch signal. The circuit is shown below.

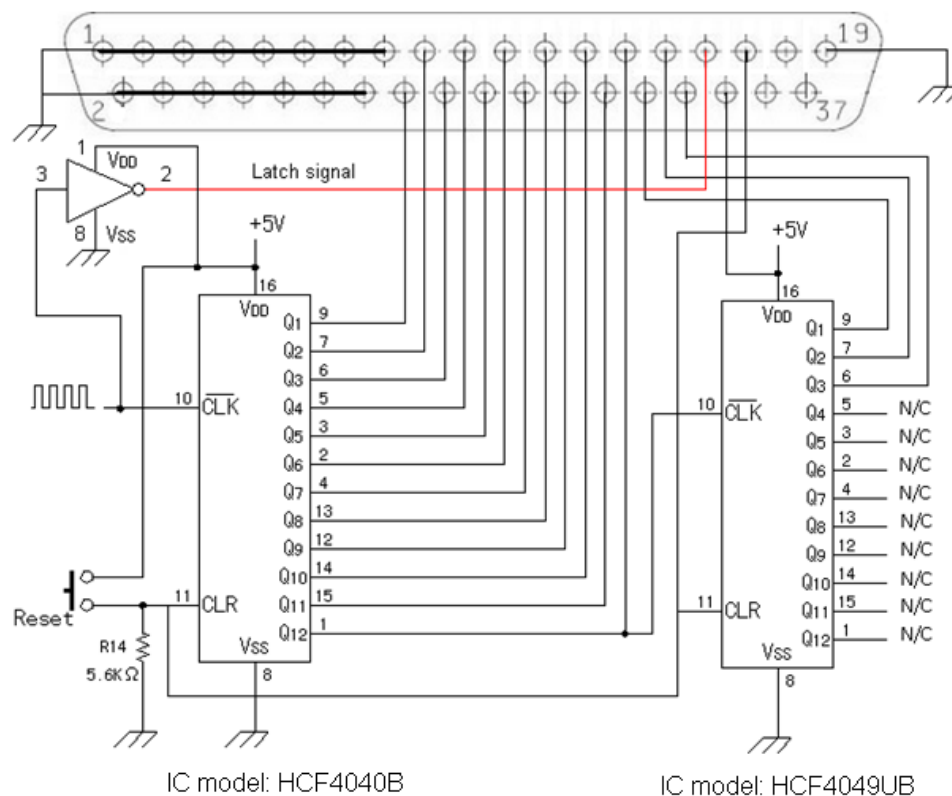


Figure A.2 Digital pulses to binary bits converter

The initial frequency of 80.07 MHz is started after sending 5240 pulses. Then 1024 pulses need to be sent to achieve a driver frequency scanning up to 95.70 MHz. This results in a wavelength that is scanned from 683.42 to 590.98 nm. The pulse generation code is shown as below. The NI-DAQ model is PCI-6221, the pulse train is output from pin 2.

```
void SENDPULSES(long lNumPulses, long lNumFrequency, float64
WaitTime)

{
    /******
    // DAQmx Configure Code
    /******
    DAQmxCreateTask("", &taskHandle);
    DAQmxCreateCOPulseChanFreq(taskHandle, "Dev1/ctr0", "", DAQmx_Val_
    l_Hz, DAQmx_Val_Low, 0.0, lNumFrequency, 0.2); //Dev1/ctrl
    DAQmx_Val_High
    DAQmxCfgImplicitTiming(taskHandle, DAQmx_Val_FiniteSamps, lNumP
    ulses);
    /******
    // DAQmx Start Code
    /******
    DAQmxStartTask(taskHandle);
    DAQmxWaitUntilTaskDone(taskHandle, WaitTime);
    DAQmxStopTask(taskHandle);
    DAQmxClearTask(taskHandle);
    return 0;
}
```

A.2 WSI Prototype

A.2.1 Light Source

The light source of the WSI (see Figure A.3) consists of a focused halogen-tungsten lamp, pinhole, collimating optics and an AOTF (Gooch and Housego, TF575-250-2-12). The diffracted light from the AOTF is coupled into the optical fibre. The SLED is a light source used to stabilise the interferometer, as discussed in chapter 7. The edge-pass filter is used to combine the SLED light with the diffracted light from the AOTF. The mechanical drawings of the light source are also shown in this section.

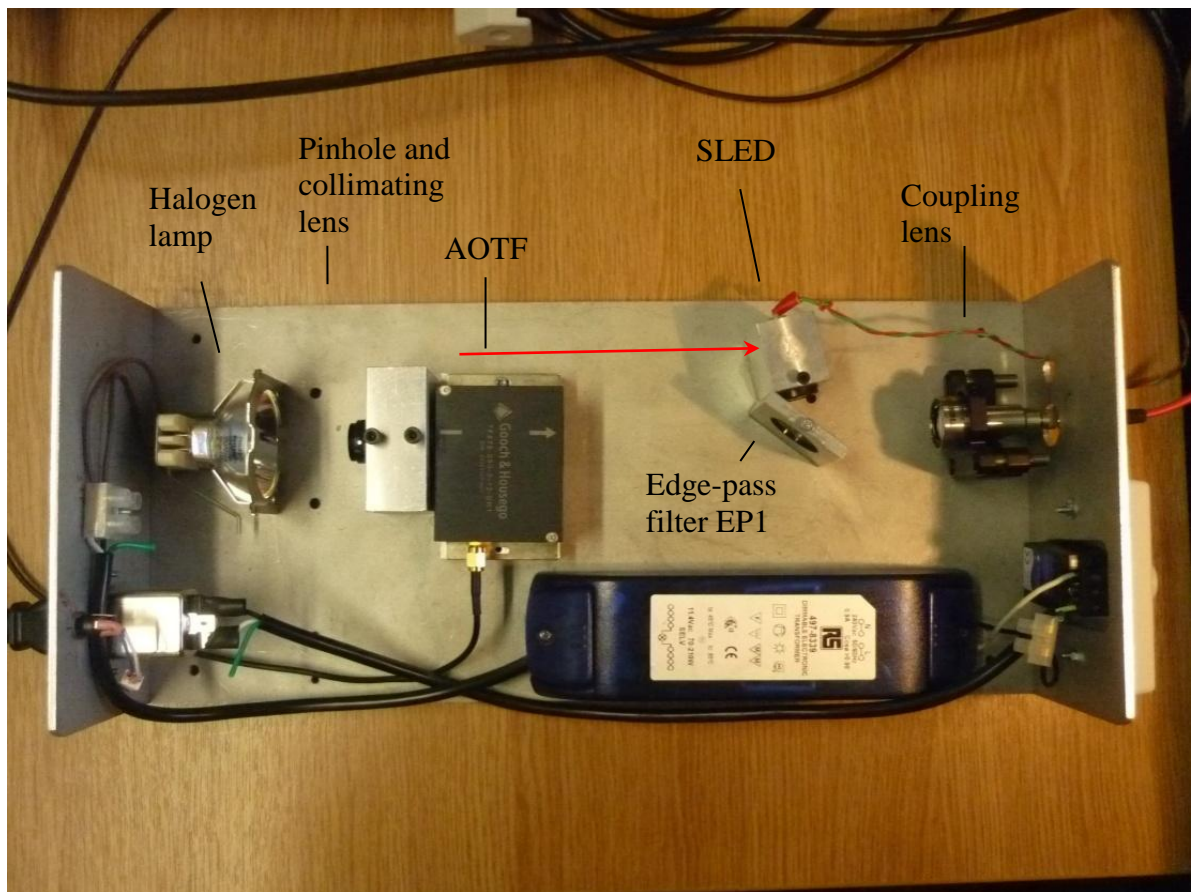
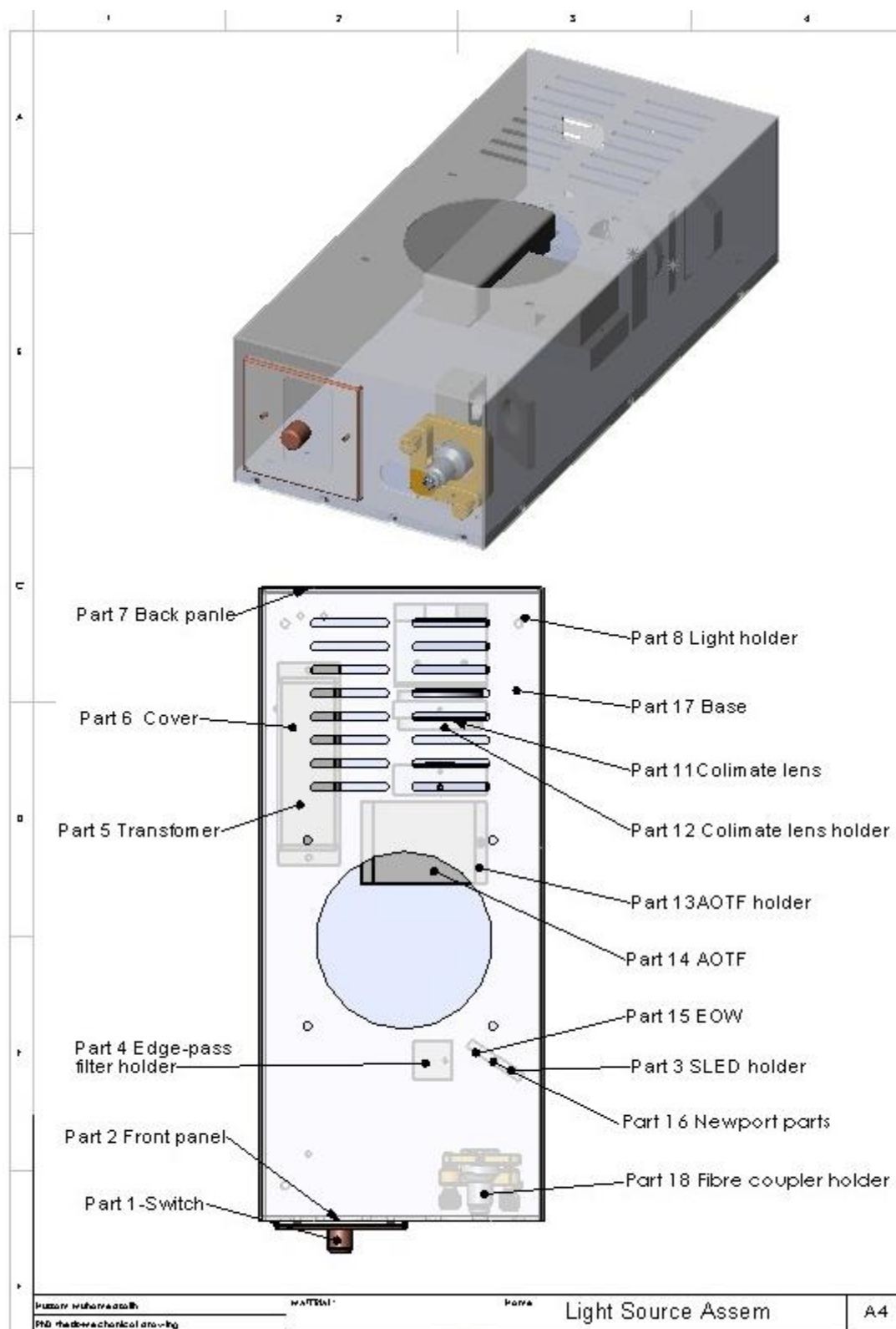
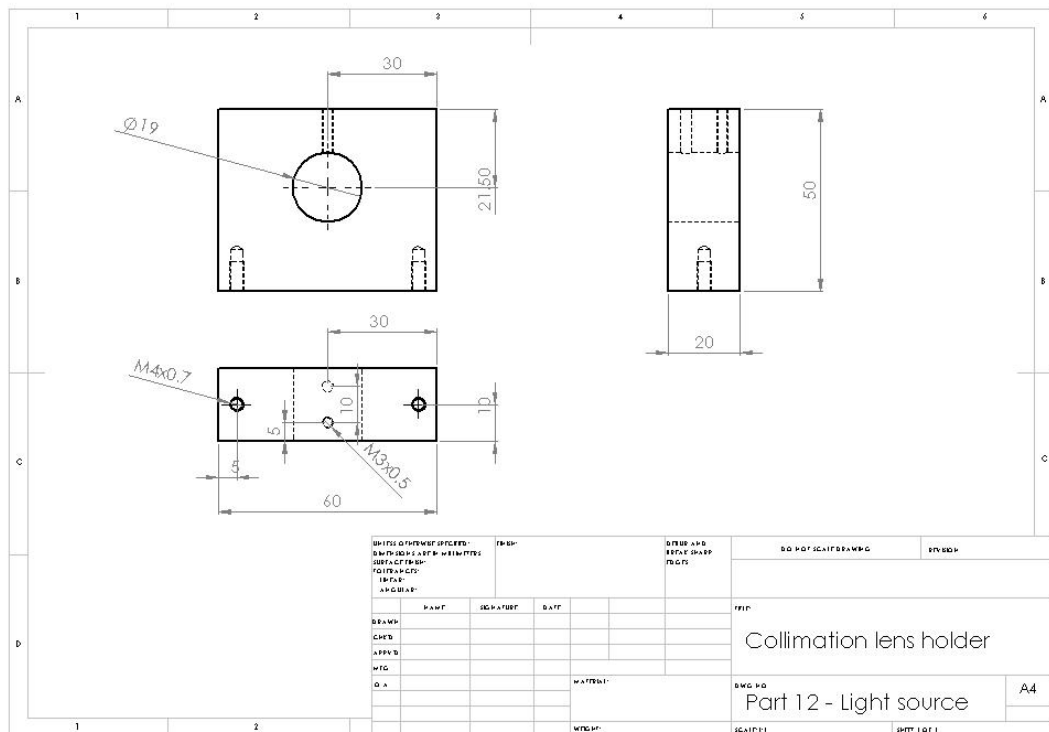
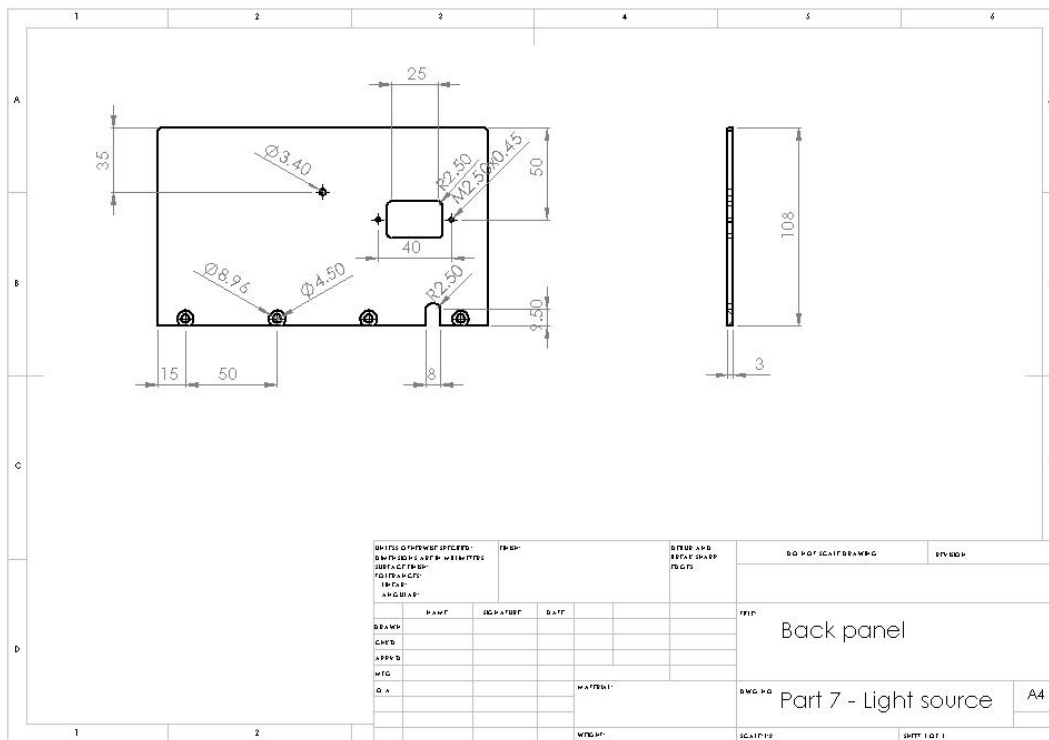


Figure A.3 Light source prototype





A.2.2 Interferometer

The interferometer of the WSI (see Figure A.4) is a Linnik configuration. The components aligned in the head of the interferometer are a beam splitter, collimating lens, imaging lens, steering mirror to fold the measurement arm and two objectives lenses which can be replaced by any matched pair of objective lenses of type of BD Plan Apo from Mitutoyo since all these objectives having the same parfocal length (95 mm). This head is fixed onto a base plate which contains also the reference mirror attached to PZT, EP2, photo-detector and CCD. The mechanical drawings are also given in this section.

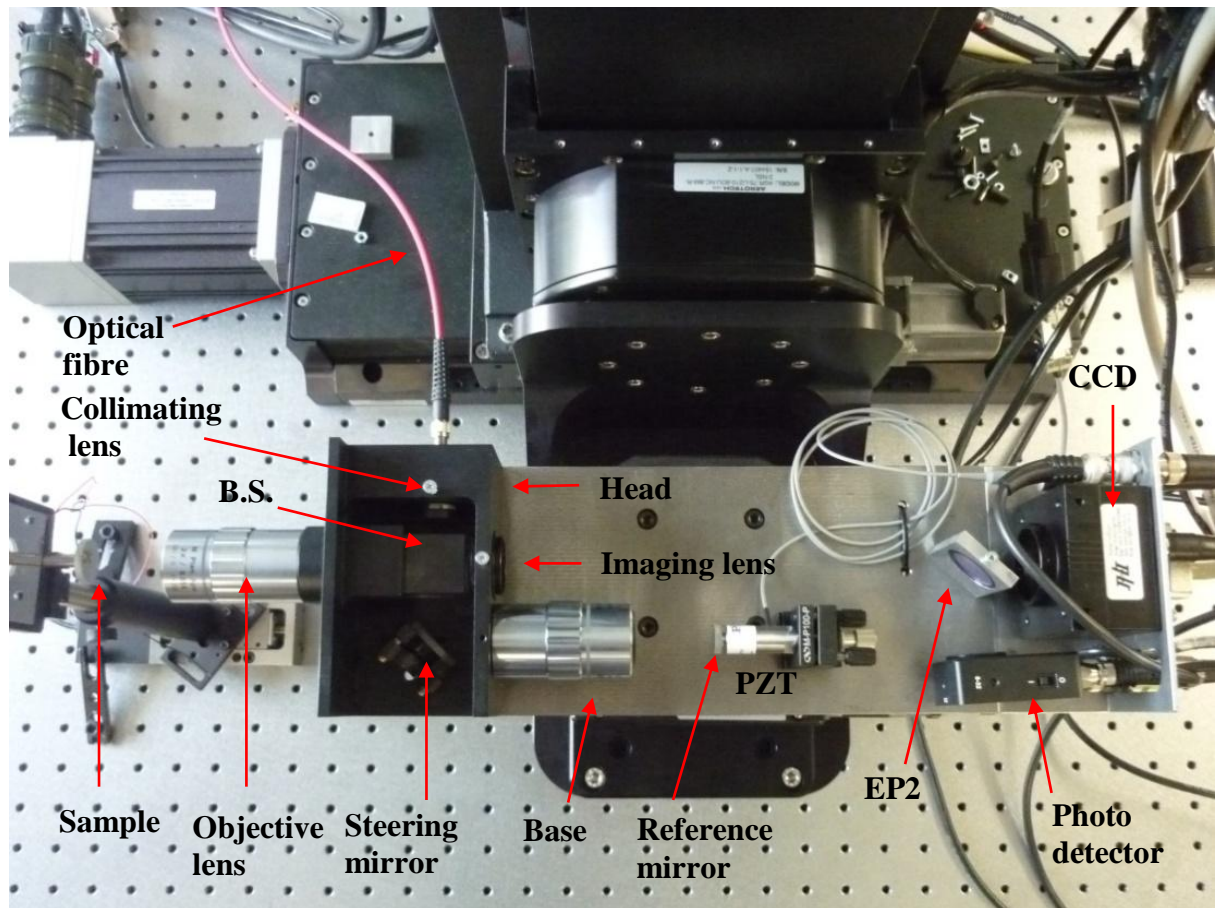
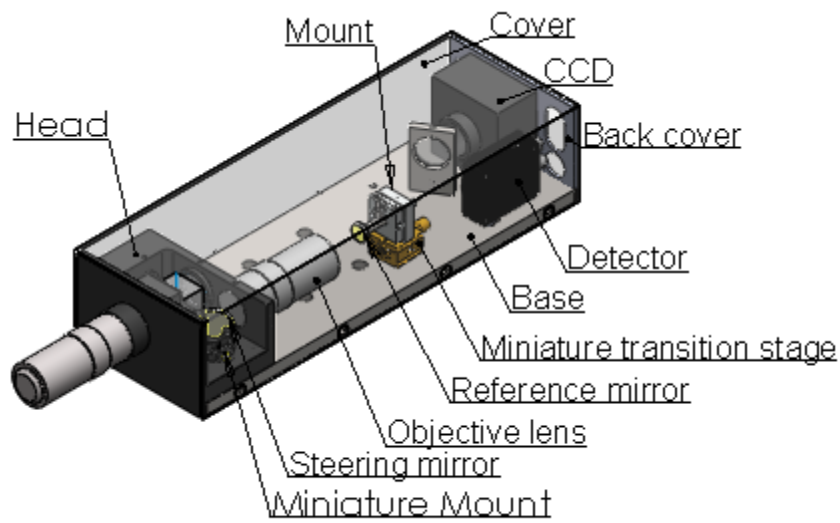
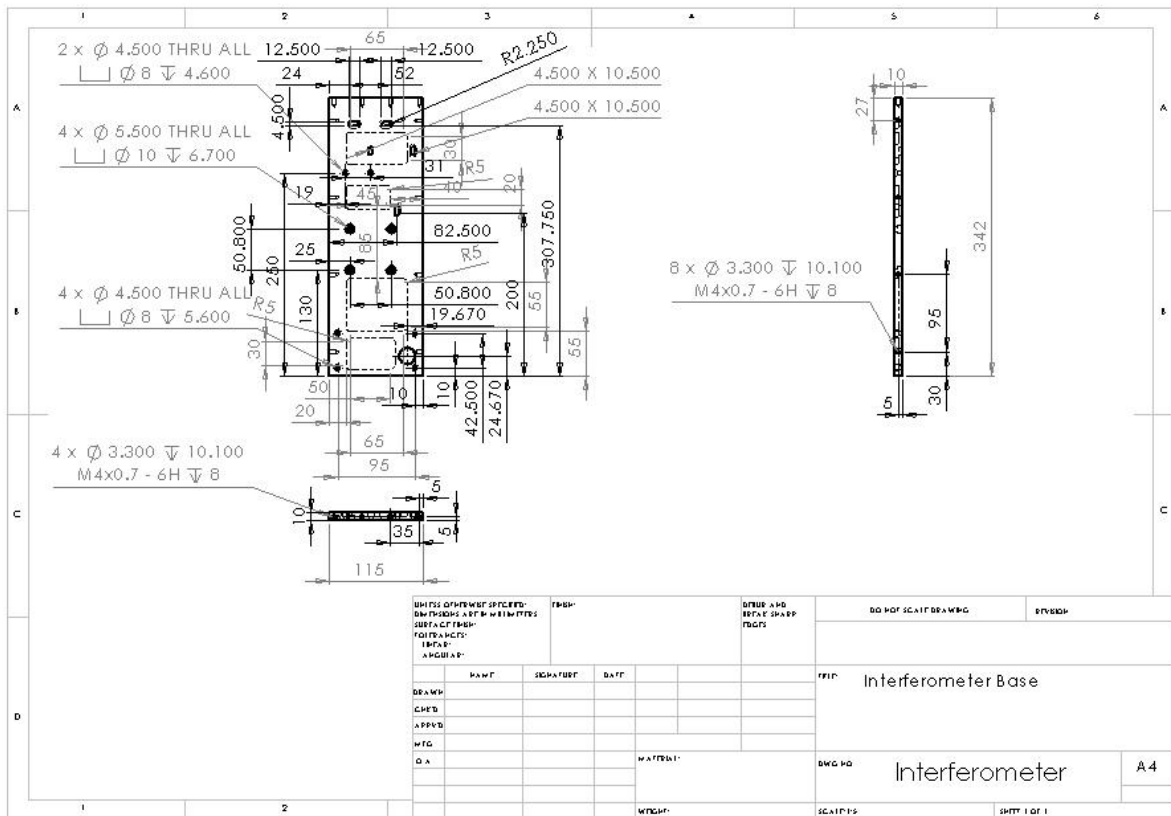
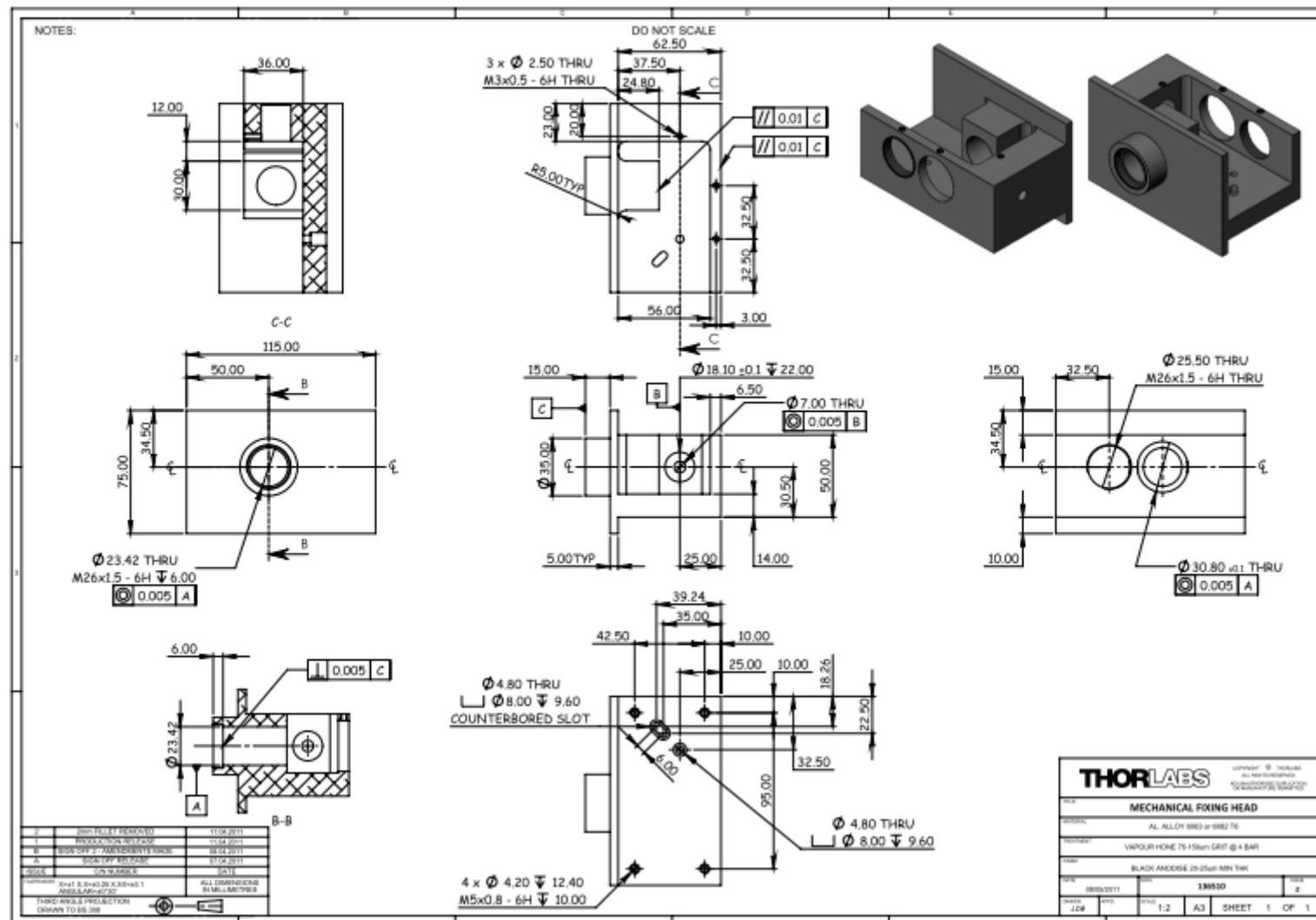


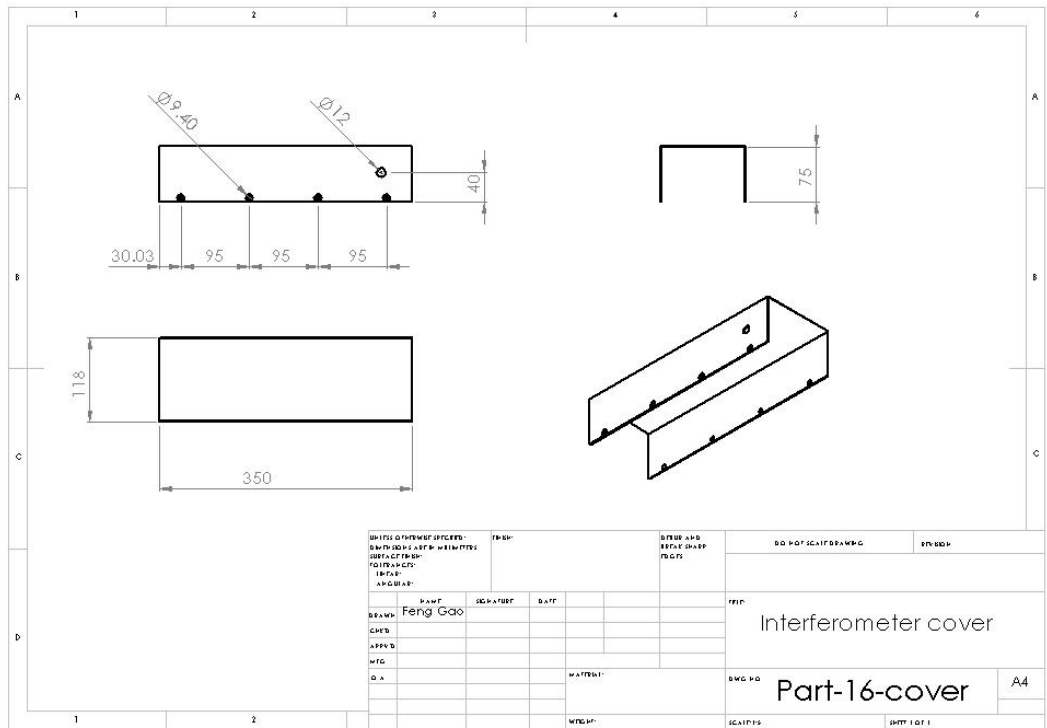
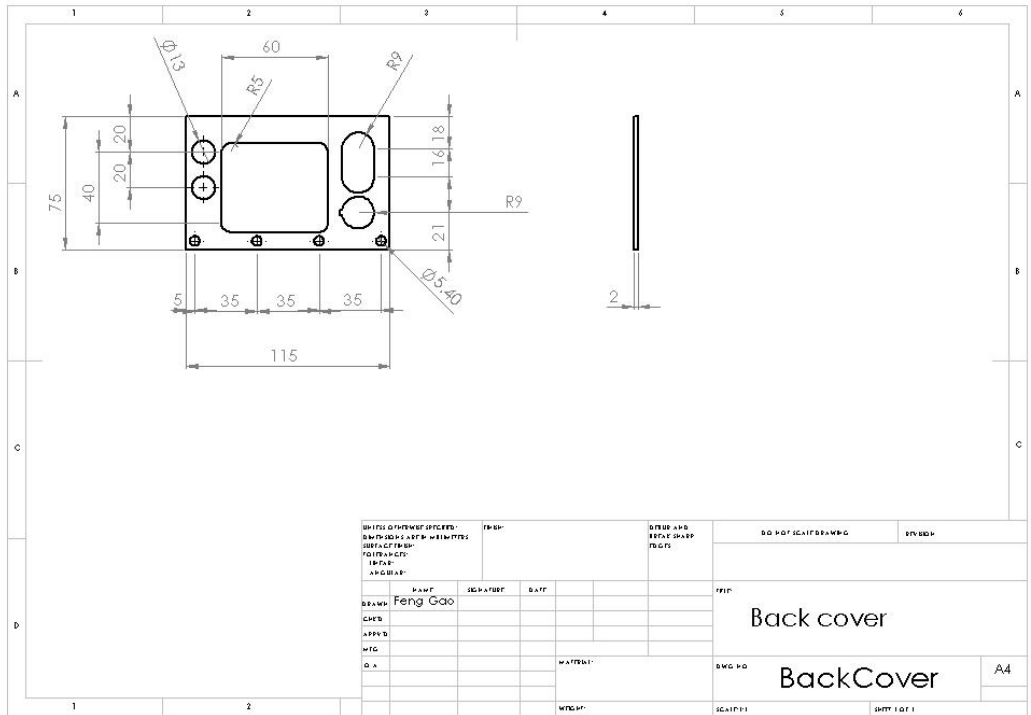
Figure A.4 WSI interferometer prototype



Interferometer







A.3 Specifications

The reflectance of the optical components, CCD responsivity and light lamp emission are shown below.

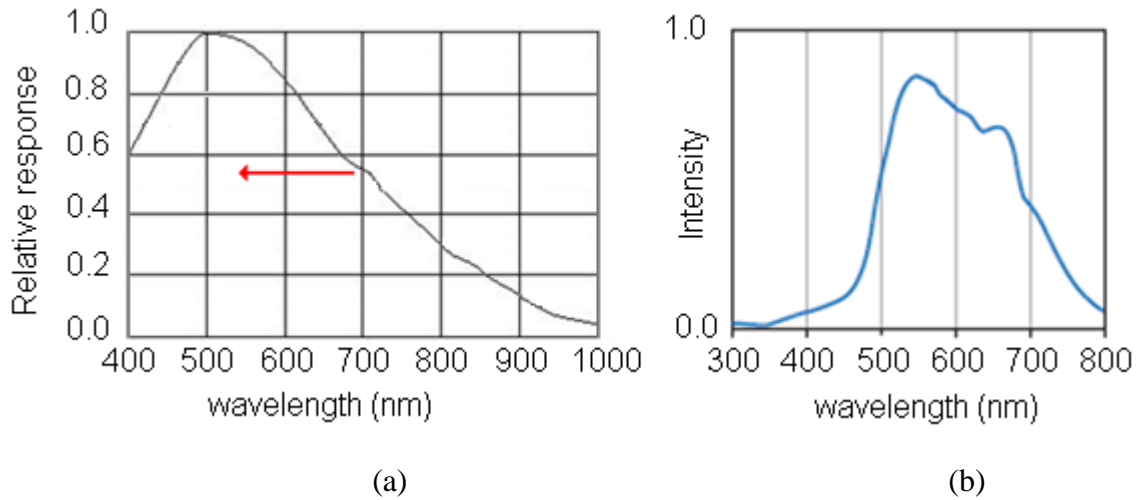


Figure A.5 (a) Response of the CCD to broadband light (b) Halogen light source emission spectrum

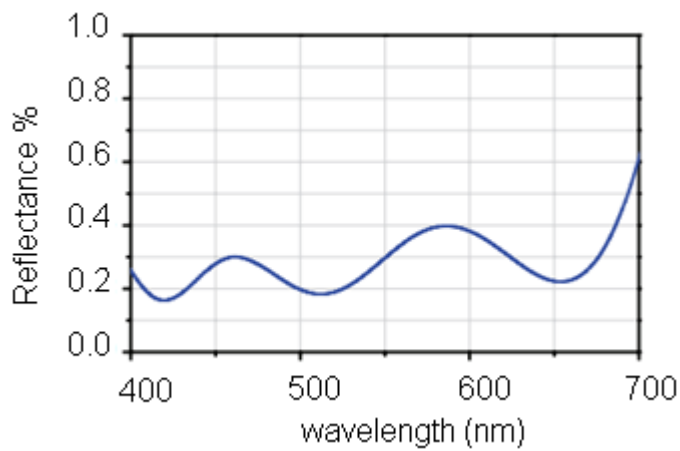


Figure A.6 Achromatic doublet visible antireflection coating

12. Appendix B

B.1 Programs of the four Algorithms: Matlab Code

This section provides the Matlab code of the described algorithms in Chapter 5.

B.1.1 Using Simple FFT

```
%Determine Surface height using Simple FFT
clc
clear;

FrameNumber=256;
m=1;
% Read the capture Data into 3D matrix (bdouble)
for i=1000:(1000+(FrameNumber-1))
path=strcat('D:\measurement
Data\DifferentSamples_Dec2011\4.707um\Stablise\6\ ',int2str(i),'.bmp');%D:\measur
ement Data\MultiStepswithSlopSample\OneFringe\
aoriginal=imread(path);
pxs=1;
pys=1;
pxe=480;
pye=640;
pxl=pxe-pxs+1;
pyl=pye-pys+1;
bdouble(m,1:pxl,1:pyl)=aoriginal(pxs:pxe,pys:pye);
    m=m+1;
end

LambdaS=683.42;      % start wavelength (nm)
LambdaE=590.98;      % End wavelength (nm)

LambdaStep=((1/LambdaE)-(1/LambdaS))/(FrameNumber-1);
sum=0;
for i=1:(FrameNumber)
Kv(i)=(1/LambdaS)+sum;      % Determine the reciprocal of wavelength assuming the
scanning is linear
lamda(i)=1/Kv(i);
sum=sum+LambdaStep;
end

forpx=1:pxl
forpy=1:pyl

yt1=double(bdouble(1:(m-1),px,py));      %      (NVIDIA1) gathering the pixel
values; once at a time
Signalfft=(1/FrameNumber)*(fft(yt1));      %      (NVIDIA2) find the fft
FFTABs=abs(Signalfft);      %      (3) find the abs value of fft, plot
it if you want to see the peak position
interval=0;      %      (4) initial position of the peak
[maximum,interval]=max(FFTABs(3:length(FFTABs)/2)); % (5)find the peak poisiton,
the peak is shifted by 2 because I want to avoid the DC value
```

```

if ((maximum/FFTABs(interval+4)) >= 1)&& (maximum/FFTABs(interval)>= 1)) %
(6) monitor condition for bad data (i.e. same concept of S/N ratio)
interval=interval+2; % (7) Determine the actual peak position. you will see
that the position of the peak is less than its actual position by one because I
have started from 2
Height=(interval-1)*((LambdaS*LambdaE/(LambdaS-LambdaE))/2); % (8)
Determine the pixel height
ArealHeight(px,py)=Height; % (9) Store the pixel height in areal matrix
else
ArealHeight(px,py)=nan; % (10) For bad data
end

end
end

data=ArealHeight/1000;% transfer the data to um scale

WriteSDF('D:\measurement Data\11April2012\4.7FFT_ONLY.sdf', data,
[2.9411,2.9411,1.0e-6]); % write the height information as SDF file

```

B.1.2 Using Fitted FFT

```

%Determine Surface height using parabola fitted FFT
clc
clear;
FrameNumber=256;
m=1;
% Read the capture Data into 3D matrix (bdouble)
for i=1000:(1000+(FrameNumber-1))
path=strcat('D:\measurement
Data\DifferentSamples_Dec2011\4.707um\Stablise\6\ ',int2str(i),'.bmp');%D:\measur
ement Data\MultiStepswithSlopSample\OneFringe\
aoriginal=imread(path);
pxs=1;
pys=1;
pxe=480;
pye=640;
pxl=pxe-pxs+1;
pyl=pye-pys+1;
bdouble(m,1:pxl,1:pyl)=aoriginal(pxs:pxe,pys:pye);
m=m+1;
end
LambdaS=683.42; % Start wavelength (nm)
LambdaE=590.98; % End wavelength (nm)

LambdaStep=((1/LambdaE)-(1/LambdaS))/(FrameNumber-1);
sum=0;
for i=1:(FrameNumber)
Kv(i)=(1/LambdaS)+sum; % Determine the reciprocal of wavelength assuming the
scanning is linear
lamda(i)=1/Kv(i);
sum=sum+LambdaStep;
end
for px=1:pxl
for py=1:pyl
yt1=double(bdouble(1:(m-1),px,py)); % (NVIDIA1) gathering the pixel values;
once at a time
Signalfft=(1/FrameNumber)*(fft(yt1)); % (NVIDIA2) find the fft

```

```

FFTTABS=abs(Signalfft); % (3) find the abs value of fft, plot it
if you want to see the peak position
interval=0; % (4) initial position of the peak
[maximum,interval]=max(FFTTABS(3:length(FFTTABS)/2)); % (5) find the peak position,
the peak is shifted by 2 because I want to avoid the DC value
if ((maximum/FFTTABS(interval+4)) >= 1) && (maximum/FFTTABS(interval)>= 1) % (6)
monitor condition for bad data (i.e. same concept of S/N ratio)
interval=interval+2; % (7) Determine the actual peak position
%---2nd order polynomial fitting-----
SignalNeedFit=FFTTABS(interval-1:interval+1); % (8) Extract three points
around the peak
SampleNo=[interval-1 interval interval+1]; % (9) Find the three points
interval
ParabolaFit=polyfit(SampleNo,SignalNeedFit',2); % (10) Fit the Data
intervalFraction=abs(ParabolaFit(2)/(2*ParabolaFit(1))); % (11) Find best
estimated peak from 1st derivative
Height=(intervalFraction-1)*((LambdaS*LambdaE/(LambdaS-LambdaE))/2); % (12)
Find the pixel height
ArealHeight(px,py)=Height; % (13) Store the pixel height in Areal matrix
else
ArealHeight(px,py)=nan; % (14) For bad data
end
end
px
end
data=ArealHeight/1000; % transfer the data to um scale
WriteSDF('D:\measurement Data\11April2012\4.7FFT_ONLY.sdf', data,
[2.9411,2.9411,1.0e-6]); % write the height information as SDF file

```

B.1.3 Fringe Analysis using FFT

```

%Fringe Analysis using FFT
clc
clear;
FrameNumber=256;
m=1;
% Read the capture Data into 3D matrix (bdouble)
for i=1000:(1000+(FrameNumber-1))
path=strcat('D:\measurement
Data\DifferentSamples_Dec2011\4.707um\Stablise\6\',int2str(i),'.bmp'); %D:\measur
ement Data\MultiStepswithSlopSample\OneFringe\
aoriginal=imread(path);
pxs=1;
pys=1;
pxe=480;
pye=640;
pxl=pxe-pxs+1;
pyl=pye-pys+1;
bdouble(m,1:pxl,1:pyl)=aoriginal(pxs:pxe,pys:pye);
m=m+1;
end

LambdaS=683.42; % Start wavelength (nm)
LambdaE=590.98; % End wavelength (nm)

LambdaStep=((1/LambdaE)-(1/LambdaS))/(FrameNumber-1);
sum=0;
for i=1:(FrameNumber)

```



```

Kv(i)=(1/LambdaS)+sum;      % Determine the reciprocal of wavelength assuming the
scanning is linear
lamda(i)=1/Kv(i);
sum=sum+LambdaStep;
end

fid                                =                                fopen('D:\measurement
Data\Calibration\April2012\2x\256\RemoveBackground_256.bin');
CalMat = fread(fid, [256], 'float');
fclose(fid);

forpx=1:pxl
forpy=1:pyl
yt1=double(bdouble(1:(m-1),px,py))./CalMat; % (NVIDIA1) gathering the pixel
values; once at a time
fftResult=(1/FrameNumber)*fft(yt1);          % (NVIDIA2) find the fft
fftFilter(1:FrameNumber)=0;                  % (3) filter out the unwanted DC and
phase conjugate
fftFilter(2:20)=fftResult(2:20);
ifftFilter=ifft(fftFilter);                  % (4) inverse fft to retrieve the
1/2Be^(phase)
phase=imag(log(ifftFilter));                  % (5) xtract the phase from the
imaginary

phase_set (NVIDIA1)=0;                        % (6) phase unwrapping
for i=1:length(phase)-1                      % (6.a) construct the Phase set
function
if abs(phase(i+1)-phase(i)) >(2*pi*0.9)
phase_set(i+1)=phase_set(i)+ 2*pi;
else
phase_set(i+1)=phase_set(i);
end
end
phase_final = phase + phase_set;              % (6.b) add the Phase set to the
determined phase
% ---Linear Fitting using least square approach
%Nummerical Methods for Engineers and Scientists 2nd Edition 2001 page 228
%by Marcel Dekker
%-----%
C=0;
D=0;
E=0;
F=0;
for(i=1:N)
    C=C+i;
    D=D+phase_final(i);
    E=E+(i*i);
    F=F+(phase_final(i)*i);
end
slop=(F-(1/N)*D*C)/(E-(1/N)*(C^2));
DC=((1/N)*D)-((slop/N)*C);
for i=1:N
    PhaseFit(i)=DC+(slop*i);
end
%-----%

ArealHeight(px,py)=(PhaseFit(FrameNumber)- PhaseFit (1))
*(1/(4*pi*(Kv(FrameNumber)-Kv(1)))); % (7) calculate the height
end
px

```

```

end
data=ArealHeight/1000;% transfer the data to um scale
WriteSDF('D:\measurementData\11April2012\4.7FFTSeq.sdf', data,
[2.9411,2.9411,1.0e-6]); % (8) write the height information as SDF file

```

B.1.4 Localise Peaks Using Convolution

```

clearall
clc
warningoff;
%Determine Surface height using parabola fitted FFT
clc
clear;
FrameNumber=256;
m=1;
% Read the capture Data into 3D matrix (bdouble)
for i=1000:(1000+(FrameNumber-1))
path=strcat('D:\measurement
Data\DifferentSamples_Dec2011\4.707um\Stablise\6\ ',int2str(i),'.bmp');%D:\measur
ement Data\MultiStepswithSlopSample\OneFringe\
aoriginal=imread(path);
pxs=1;
pys=1;
pxe=480;
pye=640;
pxl=pxe-pxs+1;
pyl=pye-pys+1;
bdouble(m,1:pxl,1:pyl)=aoriginal(pxs:pxe,pys:pye);
    m=m+1;
end

LambdaS=683.42;      % Start wavelength (nm)
LambdaE=590.98;      % End wavelength (nm)

LambdaStep=((1/LambdaE)-(1/LambdaS))/(FrameNumber-1);
sum=0;
for i=1:(FrameNumber)
Kv(i)=(1/LambdaS)+sum;      % Determine the reciprocal of wavelength assuming the
scanning is linear
lamda(i)=1/Kv(i);
sum=sum+LambdaStep;
end

filter_width=24; %Construct convolve filter
for i=1:24
if(i<13)
x(i)= -1*ones;
else
x(i)= 1*ones;
end
end

forpx=1:pxl
forpy=1:pyl

b=double(bdouble(1:FrameNumber,px,py));      %---(NVIDIA1) gathering the pixel
values; once at a time
B=conv(b,x);      %---(NVIDIA2) find the convolution

```

```

dI=B(filter_width:FrameNumber-filter_width);%---(3) remove the false data when
convolve function out of the range
n=1; %---(4) find peaks and valleys
m=1;
z=1;
for i=1:(length(dI)-1)
if ((dI(i)>=0)&(dI(i+1)<0))|((dI(i)<=0)&(dI(i+1)>0))
tzero(n)=i;
n=n+1;
if ((dI(i)>=0)&(dI(i+1)<0))
tzero_valley(m)=i;
m=m+1;
end
if ((dI(i)<=0)&(dI(i+1)>0))
tzero_peak(z)=i;
z=z+1;
end
end
end

if (tzero_peak~=0) %--- consider the peaks only
Tpeak=tzero_peak+(filter_width/2); %---(5) Determine actual peaks by shifting
the zero positions
ArealHeight(px,py) = ((m-2)/2)*(1/(Kv(Tpeak(1)))-Kv(Tpeak(length(Tpeak)))));%---
(6) Determine the pixel height
else
ArealHeight(px,py)=nan;
end

end
px
end
data=ArealHeight/1000;% transfer the data to um scale
WriteSDF('D:\measurement Data\11April2012\4.7convSeq.sdf', data,
[2.9411,2.9411,1.0e-6]); % (7) write the height information as SDF file

```

B.2 Localise Peaks

The mathematical description below shows that the peaks and valleys are shifted by magnitude equal to half of the convolution function $f(x)$. The following assumptions have been made.

- The sinusoidal term of the fringe pattern is $h(t) = b \cos[2\pi t / T]$ (B.1)

- The filter function is $f(t) = \begin{cases} -1 & 0 < t < B \\ 1 & B < t < 2B \end{cases}$ (B.2)

- The convolution, $h(t) * g(t) = \int_0^B h(t-\tau)x(\tau)d\tau$ (B.3)

where b is the amplitude of the interference contrast variation, T is the period of the interference pattern and B is half of the filter width. By substituting equations B.1 and B.2 into equation B.3, the following equation is derived.

$$\begin{aligned}
&= \int_0^B h(t-\tau)x(\tau)d\tau + \int_B^{2B} h(t-\tau)x(\tau)d\tau \\
&= -\int_0^B b \cos[2\pi(t-\tau)/T]d\tau + \int_B^{2B} b \cos(2\pi(t-\tau)/T)d\tau \\
&= -\frac{bT}{2\pi} \sin[2\pi(t-\tau)/T] \Big|_0^B + \frac{bT}{2\pi} \sin[2\pi(t-\tau)/T] \Big|_B^{2B} \\
&= \frac{bT}{2\pi} \sin(2\pi t/T) - 2 \sin(2\pi(t-B)/T) + \sin(2\pi(t-2B)/T) \\
&= \frac{bT}{\pi} \sin(2\pi(t-B)/T) \cos(2\pi B/T) - \sin(2\pi(t-B)/T) \\
&= \frac{bT}{\pi} \cos(2\pi B/T) - 1 \sin(2\pi(t-B)/T) \tag{B.4}
\end{aligned}$$

The peak locations can be determined when the convolution output equal to zero, and this is possible if and only if $t=B$ or $t=T/2+B$. Therefore, the following equation can be considered

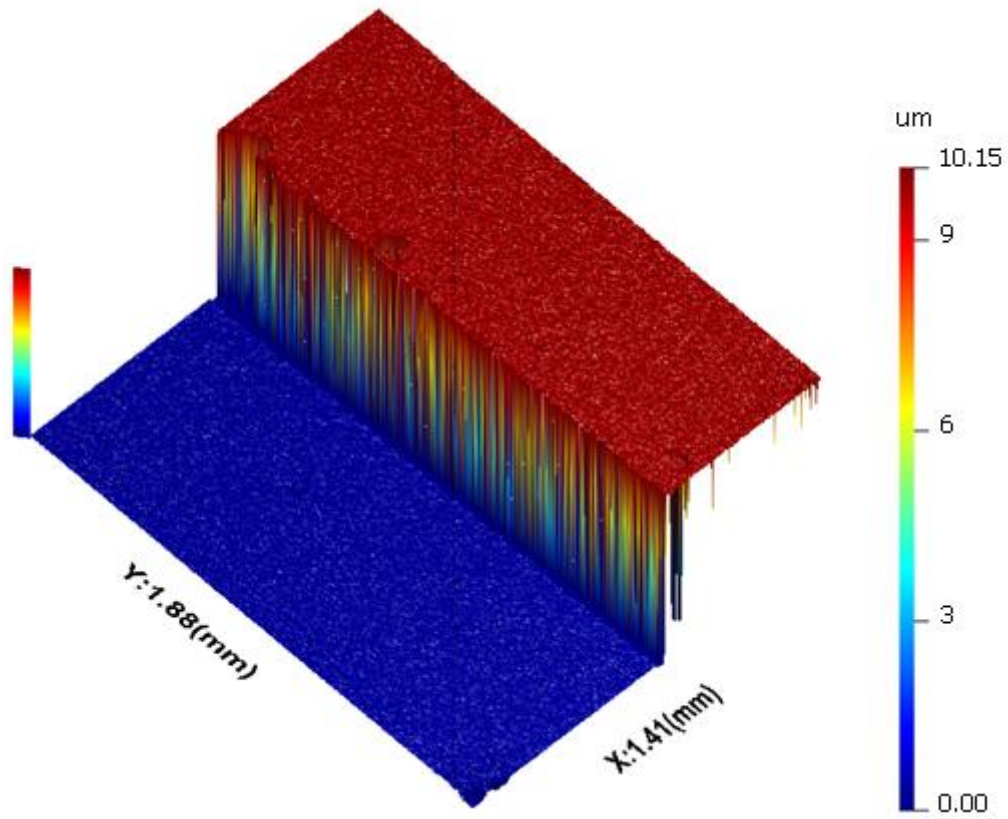
$$T_{peak} = T_{zero} - B \tag{B.5}$$

where T_{peak} is the actual location of the peaks or valleys, T_{zero} is the zero position that refers to peaks or valleys.

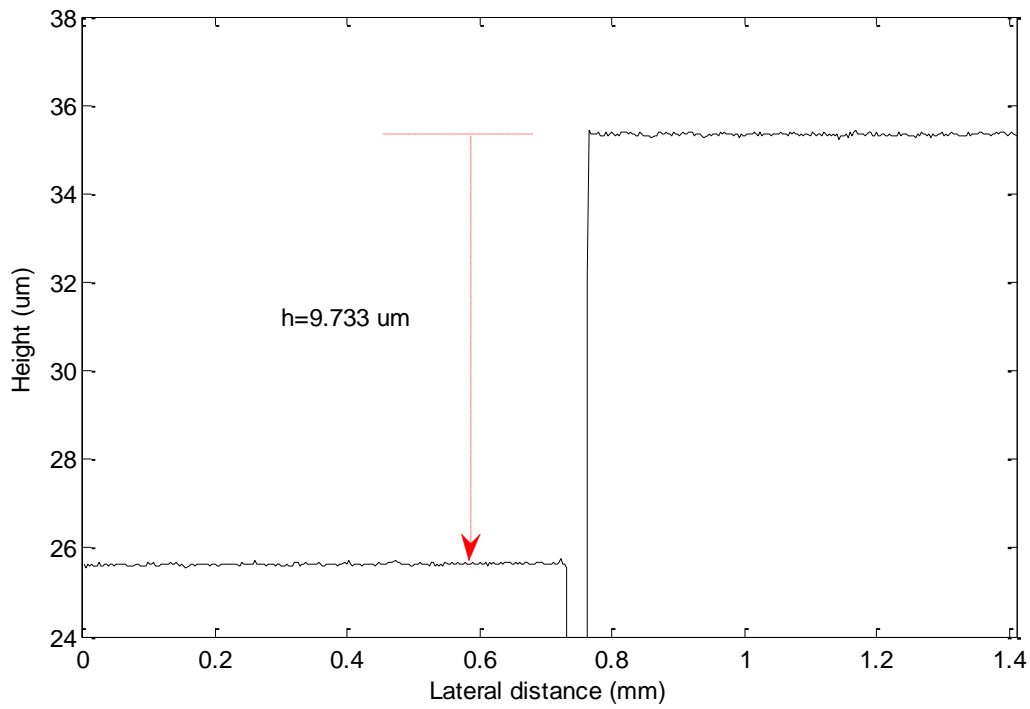
B.3 Measurement of 9.758μm Wyko Sample

The sample is a 9.758μm standard step height obtained from Wyko. This sample has been measured using WSI system and fringe analysis using algorithm C (analyse the fringe using Fourier transform). The objective lens was 2X magnification with 0.055 NA. The capturing process was synchronised with the driving frequency of the AOTF. The scanned range was 683.42 nm to 590.98 nm. Each captured frame was taken when the wavelength is shifted approximately 0.36 nm. The total number of captured frames was 256, in 2.62 second.

The step height is found by determining the difference between the two mean of upper and lower surfaces heights. The measured step height is found to be $9.733\text{ }\mu\text{m}$ (see figure B.1). Thus the error was $0.025\text{ }\mu\text{m}$. This error is likely to be due to the measurement resolution limitation.



(a)

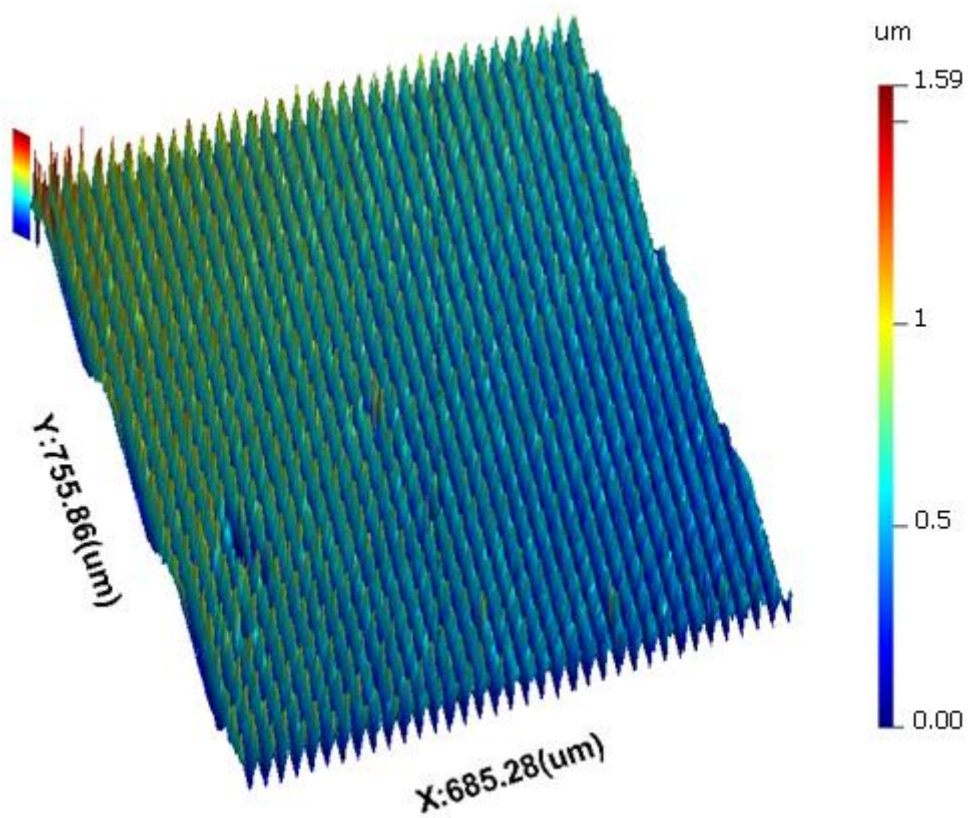


(b)

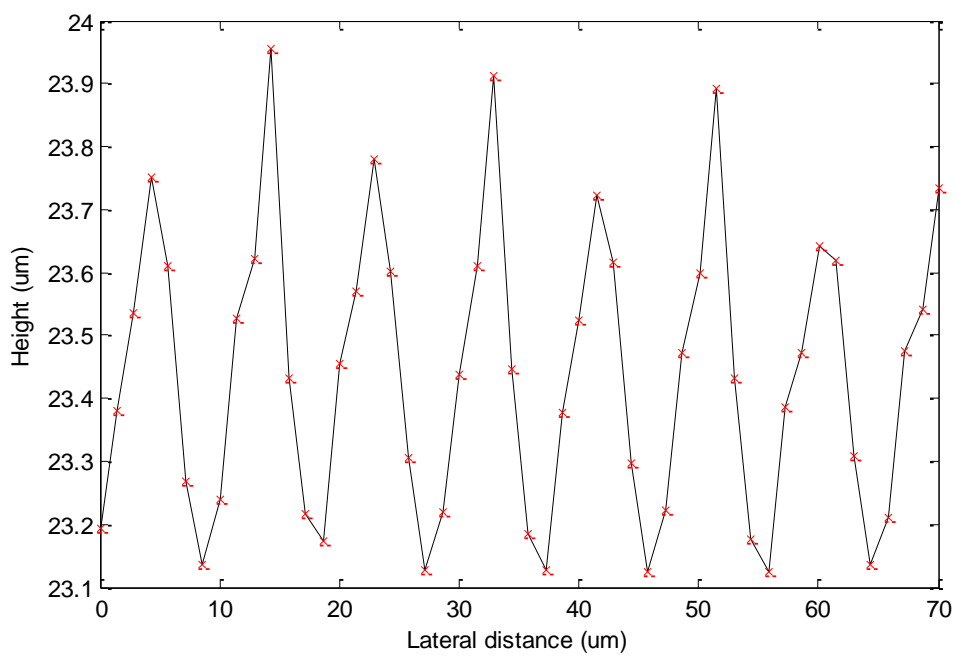
Figure B.1 Measurement of 9.758 μm step sample (a) Areal topography (b) cross-profile

B.4 Measurement of Sine Wave Sample

The sample is not a calibration standard, but has been measured by a commercial white light interferometer (Taylor Hobson, Talysurf CCI-6000). The measured wave amplitude and wave period using the CCI are 0.61 μm and 14 μm respectively. The same procedure described above was repeated using the WSI with a sample frame of 257 x 233 pixels. The amplitude and period of the wave are found to be 0.781 μm and 10.01 μm respectively (see figure B.2). The differences in the sine amplitudes are mainly caused by not providing sufficient number of pixels for the wave period.



(a)



(b)

Figure B.2 Measurement sine wave sample (a) areal topography (b) cross-profile

13. Appendix C

Appendix C.1 Data Arrangement

```
1. //Allocate Data matrix in host memory (i.e. CPU main memory)
   Note: FH=Frame height, FW=Frame width, F=Total frame number,
   row= row number in a frame, col=column number
2. BuffElement=0; //Index of the buffer matrix
3. for(int FramCounter=0; FramCounter<TotalFrameNumber;
   FramCounter++){
4.   for(row=0; row <FH; row++){
5.     for(col=0; col <FW; col++){
6.       DataElement=(row*FW*F)+(col*F)
           +FramCounter; //Index of the Data matrix
7.       DataHost[DataElement].x=(float) (BufData[BuffElement])
8.       BuffElement=BuffElement+1;}}
```

Appendix C.2 Memory Management and Transfer

```
1. cudaMalloc((void**) &DataDevice, mSize); //allocate a memory. The
   mSize is the size of the allocated array
2. cudaMemcpy(DataDevice, DataHost, mSize, cudaMemcpyHostToDevice); //Copy
   y memory from the Host to the Device
```

Appendix C.3 Thread Organisation

```
1. dim3 dimBlock(16,16); //Declare the threads in each block
2. dim3 dimGrid(40,30); //Declare the block in each Grid
```

Appendix C.4 Main Program Body (Kernel Invocation and Execution)

```
1. // Allocate memory spaces for DataDevice, PhaseDevice, PhaseSet,
   BkgrdIntDevice, KvDevice
2. // Copy the arranged frames Data and wavelength matrix from host to
   Device (i.e. copy DataHost and Kvinto DataDevice and KvDevice
   respectively)
3. RemoveBckgrndInt<<<dimGrid, dimBlock>>>>(DataDevice,
   BkgrdIntDevice, Fw, FH, F);
4. cufftHandle plan;
5. cufftPlan1d(&plan, F, CUFFT_C2C, FW*FH); //Create a plan for FFT
6. cufftExecC2C(plan, (cufftComplex*) DataDevice, cufftComplex*) DataDevic
   e, CUFFT_FORWARD); //Perform Forward FFT
7. FilteringData<<<dimGrid, dimBlock>>>>(DataDevice, Fw, FH, F); // Launch
   Filtering kernel
8. cufftExecC2C(plan, (cufftComplex *) DataDevice, (cufftComplex *)
   DataDevice, CUFFT_INVERSE);
9. cufftDestroy(plan);
10. DeterminePhase<<<dimGrid, dimBlock>>>>(DataDevice, PhaseDevice, Fw
   , FH, F);
```



```

11.      ImplementPhaseSet<<<dimGrid,dimBlock>>>(PhaseSetDevice,
      PhaseDevice,Fw,FH,F);
12.      CorrectPhase<<<dimGrid,dimBlock>>>(PhaseSetDevice,
      PhaseDevice,Fw,FH,F);
13.      LeastSquareFitting<<<dimGrid,dimBlock>>>(phase_cont_d,FrameWid
      th,HeightDivider,lNumImage);
14.      CalculateSurfaceHeight<<<dimGrid,dimBlock>>>( PhaseDevice,
      HeightDevice,KvDevice,Fw,FH,F);

```

Appendix C.5 Kernel of Removing the background intensity

```

__global__      void RemoveBckgrndInt (Complex      *DataDevice, float
*BkgrdIntDevice, long FW, long FH, long F) {
1.  int bx=blockIdx.x; int by=blockIdx.y; //2D Block ID
2.  int tx=threadIdx.x; int ty=threadIdx.y; //2D thread ID
3.  int row = BlockSize *by+ty; //identify the row of Data Segement
4.  int col = BlockSize *bx+tx; // identify the column of Data Segement
5.  long DataDeviceElement;
6.  DataDeviceElement=(row*FW*F)+(col+F); // identify the offset of Data
    Segement
7.  for(int i=0; i<F; i++){
8.  DataDevice[DataDeviceElement+i].x=DataDevice[DataDeviceElement+i].x
    /BkgrdIntDevice; // Remove the background intensity
9.  DataDevice[DataDeviceElement+i].y=0;}}
10.      __syncthreads();}

```

Appendix C.6 Kernel of Data Filtration

```

__global__      void FilteringData (Complex *DataDevice, long FW, long
FH, long F) {
1.  int bx=blockIdx.x; int by=blockIdx.y; //2D Block ID
2.  int tx=threadIdx.x; int ty=threadIdx.y; //2D thread ID
3.  int row = BlockSize *by+ty; //identify the row of Data Segement
4.  int col = BlockSize *bx+tx; // identify the column of Data Segement
5.  long DataDeviceElement;
6.  DataDeviceElement=(row*FW*F)+(col+F); // identify the offset of Data
    Segement
7.  for(int i=0; i<3; i++){
8.  DataDevice[DataDeviceElement+i].x=0;
9.  DataDevice[DataDeviceElement+i].y=0;}
10.      for (int i=20; i<F; i++){
11.      DataDevice[DataDeviceElement+i].x=0;
12.      DataDevice[DataDeviceElement+i].y=0;}
13.      __syncthreads();}

```

Appendix C.7 Kernel of Determining the Phase Shift

```

__global__      void DeterminePhase (Complex *DataDevice, float
*PhaseDevice, long FW, long FH, long FH) {
1.  int bx=blockIdx.x; int by=blockIdx.y; //2D Block ID
2.  int tx=threadIdx.x; int ty=threadIdx.y; //2D thread ID
3.  int row = BlockSize *by+ty; //identify the row of Data Segement

```

```

4. int col = BlockSize *bx+tx; // identify the column ofData Segement
5. long PhaseDeviceElement;
6. PhaseDeviceElement=(row*FW*F)+(col+F); // identify the offset of
   Data Segement
7. for(int i=0; i<F; i++){
8. PhaseDevice[PhaseDeviceElement+i]=atan2(DataDevice[PhaseDeviceEleme
   nt+i].y,DataDevice[PhaseDeviceElement+i].x);}
9. __syncthreads();}

```

Appendix C.8 Kernel of Implementing the Phase Offset Distribution

```

__global__ void ImplementPhaseSet(float *PhaseSetDevice,float
*PhaseDevice, long FW, long FH, longF){
1. int bx=blockIdx.x; int by=blockIdx.y;//2D Block ID
2. int tx=threadIdx.x; int ty=threadIdx.y;//2D thread ID
3. int row = BlockSize *by+ty; //identify the row of Data Segement
4. int col = BlockSize *bx+tx; // identify the column ofData Segement
5. long PhaseElement;
6. PhaseElement =(row*FW*F)+ col*F identify the offset of Data
   Segement
7. PhaseSetDevice[PhaseElement+0]=0;
8. for(int i=0; i<(F-1); i++){
9. PhaseSetDevice[PhaseElement+i+1]=PhaseSetDevice[PhaseElement+i];
10.     if (abs(PhaseDevice[PhaseElement+i+1]-
    PhaseDevice[PhaseElement+i]>2*pi*0.9){
11.         PhaseSetDevice[PhaseElement+i+1]=
    PhaseSetDevice[PhaseElement+i]+2*pi;// construct the 2 pi offset
12.     }__syncthreads();// Synchronize the if statement
13.     }__syncthreads();} // Synchronize the for statement

```

Appendix C.9 Kernel of Correcting the Phase Distribution

```

__global__ void CorrectPhase(float *PhaseSetDevice,float *PhaseDevice,
long FW, long FH, longF){
1. int bx=blockIdx.x; int by=blockIdx.y;//2D Block ID
2. int tx=threadIdx.x; int ty=threadIdx.y;//2D thread ID
3. int row = BlockSize *by+ty; //identify the row of Data Segement
4. int col = BlockSize *bx+tx; // identify the column ofData Segement
5. long PhaseElement;
6. PhaseElement =(row*FW*F)+col*F// identify the offset of Data
   Segement
7. for(int i=0; i<F; i++){
8. PhaseDevice[PhaseElement+i]=
    PhaseDevice[PhaseElement+i]+PhaseSetDevice[PhaseElement+i]; // add
    the 2 pi offset to the phase distribution
9. __syncthreads();} // Synchronize the for statement

```

Appendix C.10 Kernel of Fitting the corrected phase

```

__global__ void LeastSquareFitting(float *phase_cont_d,long FrameWidth,
long HeightDivider,long lNumImage){

```

```

1. int bx=blockIdx.x; int by=blockIdx.y;//2D Block ID
2. int tx=threadIdx.x; int ty=threadIdx.y;//2D thread ID
3. int row = BlockSize *by+ty; //identify the row of Data Segement
4. int col = BlockSize *bx+tx; //identify the column ofData Segement
5. long PhaseElement; float N=lNumImage;
6. PhaseElement =(row*FW*F)+col*F; //identify the offset of Data
   Segement
7. float C=0;float D=0;float E=0;float F=0;float slop;float DC;
8. for (int i=1;i<=lNumImage;i++){
9. C=C+i;
10.    D=D+phase_cont_d[element_phase+(i-1)];
11.    E=E+(i*i);
12.    F=F+(phase_cont_d[element_phase+(i-1)]*i);}
13.    __syncthreads();// Synchronize the for statement
14.    slop =(F-((1/ N)*D*C))/(E-(1/ N)*(C*C));
15.    DC=((1/ N)*D)-((slop/ N)*C);
16.    for (int i=1;i<=lNumImage;i++){
        phase_cont_d[element_phase+(i-1)]=DC+(slop*i);}
17.    __syncthreads();} // Synchronize the for statement

```

Appendix C.10 Kernel of Calculating the Surface Height

```

global__ voidCalculateSurfaceHeight(float *PhaseDevice,float
*HeightDevice, float *KvDevice, long FW, long FH, longF){
1. int bx=blockIdx.x; int by=blockIdx.y;//2D Block ID
2. int tx=threadIdx.x; int ty=threadIdx.y;//2D thread ID
3. int row = BlockSize *by+ty; //identify the row of Data Segement
4. int col = BlockSize *bx+tx; // identify the column ofData Segement
5. longHeightElement;
6. HeightElement =(row*FW*F)+col*F;// identify the offset of Data
   Segement
7. for(int i=0; i<F; i++){
8. HeightDevice [HeightElement+i]=(PhaseDevice [HeightElement +(F-1)]-
   PhaseDevice[HeightElement])/(4*pi*(KvDevice[(F-1)]-KvDevice[0]));}
9. __syncthreads();} // Synchronize the for statement

```

Appendix C.11 Write the Surface Height Data into SDF file

```

1. charVerName[9]="bBCR-1.0";//This represents the version number and
   must be placed first
   charManufacID[11]="THP-FORM ";//Manufacturer's ID
   charCreateDate[13]="040519931442";//Creation Date
   charModDate[13]="020619930931";//Modification Date
   unsignedintNumPoints=FrameWidth;// Number of points per profiles
   unsignedint NumProfiles=HeightDivider*ProcessStep;// number of
   profiles
   double Xscale=2.9411E-006;// Lateral X-Scale of 2X lens
   double Yscale=2.9411E-006;// Lateral Y-Scaleof 2X lens
   double Zscale=1.00E-009;// Vertical Z-Scale
   double Zresolution=-1;// The -1 refers to unknown

```

```

    char Compression=0;//NULL;
    char DataType=3; // The 3 refers to float type
    char CheckSum=0;//NULL;
2. FILE * SurfaceFile;
3. SurfaceFile = fopen (szFileName,"wb");
4. fwrite(VerName,1,8,SurfaceFile);
5. fwrite(ManufacID,1,10,SurfaceFile);
6. fwrite(CreateDate,1,12,SurfaceFile);
7. fwrite(ModDate,1,12,SurfaceFile);
8. fwrite(NumPoints,1,2,SurfaceFile);
9. fwrite(NumProfiles,1,2,SurfaceFile);
10.    fwrite(Xscale,1,8,SurfaceFile);
11.    fwrite(Yscale,1,8,SurfaceFile);
12.    fwrite(Zscale,1,8,SurfaceFile);
13.    fwrite(Zresolution,1,8,SurfaceFile);
14.    fwrite(Compression,1,1,SurfaceFile);
15.    fwrite(DataType,1,1,SurfaceFile);
16.    fwrite(CheckSum,1,1,SurfaceFile);
17.    ElementWritten=fwrite(SurfaceHeightHost,sizeof(float),FW*FH,Su
        rfaceFile);//Write the Surface Height information
18.    fclose (SurfaceFile);

```

Appendix C.12 The main fringe analysis by FFT using CUDA program

```

#include<cutil_inline.h>
#include<cutil.h>
#include<math.h>
#include<FFT.h>
#include<FFT_kernel.cu>

long BuffElement=0;
long      DataElement;
long      ElementWritten;
int i,j;
int      FrameCounter;
FILE * IntensityValueFile;
char Hussbuffer [50]; // delet it
FILE * pPixelFile;

extern"C"
void DeviceFFT(LPBYTE lpLinear,long FrameWidth,long FrameHeight,char*
szFileName,float* Kv, long lNumImage,long lNumPulses2, int LensX);

void DeviceFFT(LPBYTE lpLinear,long FrameWidth,long FrameHeight,char*
szFileName,float* Kv, long lNumImage, long lNumPulses2, int LensX)
{

    //-----Transfer the reciprocal of wavelength array (Kv) to GPU
Global memory -----
    float *Kv_d;
    cudaMalloc((void**)&Kv_d,lNumImage*sizeof(float)); // see page 48
the bottom note for cudaMalloc function

```

```

        cudaMemcpy(Kv_d,Kv,lNumImage*sizeof(float),cudaMemcpyHostToDevice);
//Copy wavelength values
//-----
-----

/*----The following code statments are calibration steps
----The clibrated matrix has been processed by Matlab and saved
in D drive-----*/

FILE * pCalMat; // calibration matrix

//--- Allocate memory space for the background Inensity array in
CPU
float *CalMatHost=(float*)malloc(sizeof(float)*lNumImage);//on the
host side

if (LensX==2)
{
    if (lNumImage==256)
    {
        pCalMat=fopen("D:\\measurement
Data\\Calibration\\April2012\\2x\\256\\RemoveBackground_256","rb");
    }
    elseif (lNumImage==128)
    {
        pCalMat=fopen("D:\\measurement
Data\\Calibration\\April2012\\2x\\128\\RemoveBackground_128","rb");
    }
    elseif (lNumImage==64)
    {
        pCalMat=fopen("D:\\measurement
Data\\Calibration\\April2012\\2x\\64\\RemoveBackground_64","rb");
    }
}
elseif (LensX==5)
{
    if (lNumImage==256)
    {
        pCalMat=fopen("D:\\measurement
Data\\Calibration\\April2012\\5x\\256\\RemoveBackground_256","rb");
    }
    elseif (lNumImage==128)
    {
        pCalMat=fopen("D:\\measurement
Data\\Calibration\\April2012\\5x\\128\\RemoveBackground_128","rb");
    }
    elseif (lNumImage==64)
    {
        pCalMat=fopen("D:\\measurement
Data\\Calibration\\April2012\\5x\\64\\RemoveBackground_64","rb");
    }
}
}

```

```

elseif (LensX==10)
{
    if (lNumImage==256)
    {
        pCalMat=fopen("D:\\measurement
Data\\Calibration\\April2012\\10x\\256\\RemoveBackground_256","rb");
    }
    elseif (lNumImage==128)
    {
        pCalMat=fopen("D:\\measurement
Data\\Calibration\\April2012\\10x\\128\\RemoveBackground_128","rb");
    }
    elseif (lNumImage==64)
    {
        pCalMat=fopen("D:\\measurement
Data\\Calibration\\April2012\\10x\\64\\RemoveBackground_64","rb");
    }
}

if (pCalMat!=NULL)
{
    fread(CalMatHost,sizeof(float),lNumImage,pCalMat);
    fclose(pCalMat);
}

float *CalMat_d;// discontinuous phase matrix
cudaMalloc((void**)&CalMat_d,sizeof(float)*lNumImage);
cudaMemcpy(CalMat_d,CalMatHost,sizeof(float)*lNumImage,cudaMemcpyHostToDevice);

//-----END of Calibration-----

//---Copy the BUFFER data to allocate memory BufData
byte *BufData=(byte*)malloc(lNumImage*FrameWidth*FrameHeight);
memcpy(BufData,lpLinear,lNumImage*FrameWidth*FrameHeight);//copy
the BUFFER data to allocate memory BufData
//-----

//-----Divide the Captured Data into 4 parts because of memory
limitation-----
int ProcessStep=4;
int HeightDivider=FrameHeight/ProcessStep;
int CorrectionFactor=HeightDivider/BLOCK_SIZE;
HeightDivider=CorrectionFactor*BLOCK_SIZE;
int RemainPixels=FrameHeight-HeightDivider*ProcessStep;
//-----

```

```

        unsignedint size_StepHeight_d=HeightDivider*FrameWidth; //
        unsignedint mem_size_StepHeight_d=sizeof(float)*size_StepHeight_d;
        float
*HostFinalStepHeight=(float*)malloc(sizeof(float)*HeightDivider*FrameWidth
h* ProcessStep);

        for(int
ProcessCounter=0;ProcessCounter<ProcessStep;ProcessCounter++)
        {
            //----Allocate a memory space for Data in the GPU
            Complex *data_d;
            unsignedint size_data_d=lNumImage*HeightDivider*FrameWidth; //
1D_array_matrix
            unsignedint mem_size_data_d=sizeof(Complex)* size_data_d;
            cudaMalloc((void**) &data_d,mem_size_data_d);
            //----

            //----Allocate a memory space for Phase in the GPU
            float *phase_discont_d;// discontinuous phase matrix
            unsignedint size_phase_discont=lNumImage*HeightDivider*FrameWidth;
// 1D_array_matrix
            unsignedint mem_size_phase_discont=sizeof(float)*
size_phase_discont;
            cudaMalloc((void**) &phase_discont_d,mem_size_phase_discont);
            //----

            //----Allocate a memory space for phase set and corrected phase in
the GPU
            float *phase_set_d;// discontinuous phase matrix
            cudaMalloc((void**) &phase_set_d,mem_size_phase_discont);
            float *phase_cont_d; // discontinuous phase matrix
            cudaMalloc((void**) &phase_cont_d,mem_size_phase_discont);
            //----

            //----Allocate a memory space for Surface Height in the GPU
            float *StepHeight_d;
            unsignedint size_StepHeight_d=HeightDivider*FrameWidth; //
            unsignedint mem_size_StepHeight_d=sizeof(float)*size_StepHeight_d;
            cudaMalloc((void**) &StepHeight_d,mem_size_StepHeight_d);
            //----

            //----Allocate a memory space for Surface Height in the Host
            float *HostStepHeight=(float*)malloc(mem_size_StepHeight_d);//on
the host side
            //----

            /*****
            /      Replace the Buffer information into Data matrix
            *****/

```

```

    unsignedint
size_Data=lNumImage*FrameWidth*HeightDivider;//FrameHeight*FrameWidth;
    unsignedint mem_size_Data=sizeof(Complex)*size_Data;
    Complex *Data=(Complex*)malloc(mem_size_Data);// the data that is
going to be processed
    DataElement=0;

    for (FrameCounter=0; FrameCounter<lNumImage;FrameCounter++)
    {
        BuffElement=0;
        for(i=0; i<HeightDivider; i++){
            for(j=0; j<FrameWidth; j++){

                DataElement=(i*FrameWidth*lNumImage)+(j*lNumImage)+FrameCounter;
                //var_height(j)=row number & var_width(i)= col number

                Data[DataElement].x=(float) (BufData[BuffElement+((ProcessCounter*He
ightDivider)*FrameWidth)+FrameCounter*(FrameHeight*FrameWidth)]);
                BuffElement=BuffElement+1;
            }
        }
    }
    //-----End of Data Arrangement-----

    //-----Start Slop phase Analysis using GPU-----

    //---- (NVIDIA1) Transfer arrangement frames into GPU
    cudaMemcpy(data_d,Data,mem_size_data_d,cudaMemcpyHostToDevice);
    //----

    //---- (NVIDIA2) Thread Generation
    // This condition has set because the multiplied matrix is square
and can divide over tile width
    dim3 dimBlock(BLOCK_SIZE,BLOCK_SIZE); //specify the dimensions of
each BLOCK in terms of number of thread
    dim3 dimGrid(FrameWidth/BLOCK_SIZE,HeightDivider/BLOCK_SIZE);
    //specify the dimensions of each GRID in terms of number of thread
    //----

    //--- (3) Remove the Background Intensity
    CalibrationIntensityPattern<<<dimGrid,dimBlock>>>(data_d,CalMat_d,F
rameWidth,HeightDivider,lNumImage);

    //--- (4) Apply FFT
    cufftHandle plan;
    cufftPlan1d(&plan, lNumImage, CUFFT_C2C, FrameWidth *
HeightDivider);
    cufftExecC2C(plan, (cufftComplex *)data_d, (cufftComplex *)data_d,
CUFFT_FORWARD);

    //--- (5) Find the peak of spectrum to identify the filtering
cutting edge

```



```

fftFindPeaks<<<dimGrid,dimBlock>>>(data_d,HeightDivider,FrameWidth,
lNumImage);

//--- (6) Filter the DC bias, the conjugate of the phase and noises
filtering_data_d<<<dimGrid,dimBlock>>>(data_d,FrameWidth,HeightDivi
der,lNumImage);

//--- (7) Apply inverse FFT
cufftExecC2C(plan, (cufftComplex *)data_d, (cufftComplex *)data_d,
CUFFT_INVERSE);
cufftDestroy(plan);// important step; otherwise you can not repeat
the program

//--- (8) Determine the phase
Determine_phase_discont<<<dimGrid,dimBlock>>>(data_d,phase_discont_
d,FrameWidth,HeightDivider,lNumImage);

//--- (9) Compute the 2 pi phase distrubution
Implement_phase_set<<<dimGrid,dimBlock>>>(phase_discont_d,phase_set
_d,FrameWidth,HeightDivider,lNumImage);

//--- (10) Correct the phase by adding 9 to 8
Calculate_phase_cont<<<dimGrid,dimBlock>>>(phase_cont_d,phase_disco
nt_d,phase_set_d,FrameWidth,HeightDivider,lNumImage);// adding the
discontinuous phase to the phase set function

//--- (11) Fit the phase using least square approach
LeastSquareFitting<<<dimGrid,dimBlock>>>(phase_cont_d,FrameWidth,He
ightDivider,lNumImage);

//--- (12) Find the Suface Height
Find_StepHeight<<<dimGrid,dimBlock>>>(StepHeight_d,phase_cont_d,Fra
meWidth,HeightDivider,Kv_d,lNumImage);

//--- (13) upload the result back to CPU
cudaMemcpy(HostStepHeight,StepHeight_d,mem_size_StepHeight_d,cudaMe
mcpyDeviceToHost);//FrameWidth*HeightDivider*sizeof(float)

//--- (14) Place the result in a successive manner in the surface
height data that going to written as SDF
memcpy(HostFinalStepHeight+(ProcessCounter*FrameWidth*HeightDivider
),HostStepHeight,mem_size_StepHeight_d);

//---Delet the arrays---
cudaFree(data_d);
cudaFree(phase_discont_d);
cudaFree(phase_set_d);
cudaFree(phase_cont_d);
cudaFree(StepHeight_d);
free(HostStepHeight);
free(Data);
}

```

```

//--- (15) Write the Complete Suface Height Data as SDF file

//--- (15.a) Defines the entry point for the console
application.
char SDF_ver_num[9]="bBCR-1.0"; // This step must be
wriiten first
char SDF_ManufacID[11]="THP-FORM ";
char SDF_CreateDate[13]="040520121442";
char SDF_ModDate[13]="020620120931";
unsignedint SDF_NumPoints=FrameWidth;
unsignedint
SDF_NumProfiles=HeightDivider*ProcessStep; //+RemainPixels
double SDF_Xscale=3.5300E-006;
double SDF_Yscale=3.5300E-006;
double SDF_Zscale=1.00E-009;
double SDF_Zresolution=-1;
char SDF_Compression=0; //NULL;
char SDF_DataType=3; // 3 for float 6 for long 5 for short
char SDF_CheckSum=0; //NULL;

//--- (15.b) Set the laterial scale according to the
magnification scale
if (LensX==5)
{
    SDF_Xscale=1.1905E-006;
    SDF_Yscale=1.1905E-006;
}
elseif (LensX==10)
{
    SDF_Xscale=0.6098E-006;
    SDF_Yscale=0.6098E-006;
}

elseif (LensX==50)
{
    SDF_Xscale=0.1185E-006;
    SDF_Yscale=0.1185E-006;
}
else
{
    SDF_Xscale=2.9412E-006; // The 2X magnification is the
default one
    SDF_Yscale=2.9412E-006;
}

FILE * SurfaceFile;
SurfaceFile = fopen (szFileName, "wb");

//--- (15.c) write the information onto SDF file
fwrite(SDF_ver_num,1,8,SurfaceFile);
fwrite(SDF_ManufacID,1,10,SurfaceFile);
fwrite(SDF_CreateDate,1,12,SurfaceFile);
fwrite(SDF_ModDate,1,12,SurfaceFile);

fwrite(&SDF_NumPoints,1,2,SurfaceFile);

```

```

fwrite(&SDF_NumProfiles,1,2,SurfaceFile);

fwrite(&SDF_Xscale,1,8,SurfaceFile);
fwrite(&SDF_Yscale,1,8,SurfaceFile);
fwrite(&SDF_Zscale,1,8,SurfaceFile);

fwrite(&SDF_Zresolution,1,8,SurfaceFile);
fwrite(&SDF_Compression,1,1,SurfaceFile);
fwrite(&SDF_DataType,1,1,SurfaceFile);
fwrite(&SDF_CheckSum,1,1,SurfaceFile);

ElementWritten=fwrite(HostFinalStepHeight,sizeof(float),HeightDivid
er*FrameWidth*ProcessStep,SurfaceFile);//SizeHostStepHeight

fclose (SurfaceFile);

//--- (16) Free the remain arrays
free(HostFinalStepHeight);
free(Kv);
cudaFree(Kv_d);
free(BufData);
free(CalMatHost);
cudaFree(CalMat_d);

}

//-----END-----
-----

```

14. Appendix D

D.1 Implementation of PI Controller

A PI controller has been implemented to control the PZT (see figure D.1). The feedback of the controller is obtained from the interferometer output. The detected signal is manipulated by amplifying the voltage and filter out the noises. Then it is fed into PI controller to produce an electric signal for the PZT in order to produce a mechanical control action.

The electric power consists of DC to DC converter that converts 28 V, obtained from an AC/DC transformer, into ± 15 V. The ICs that are used in this circuit are driven by ± 15 V.

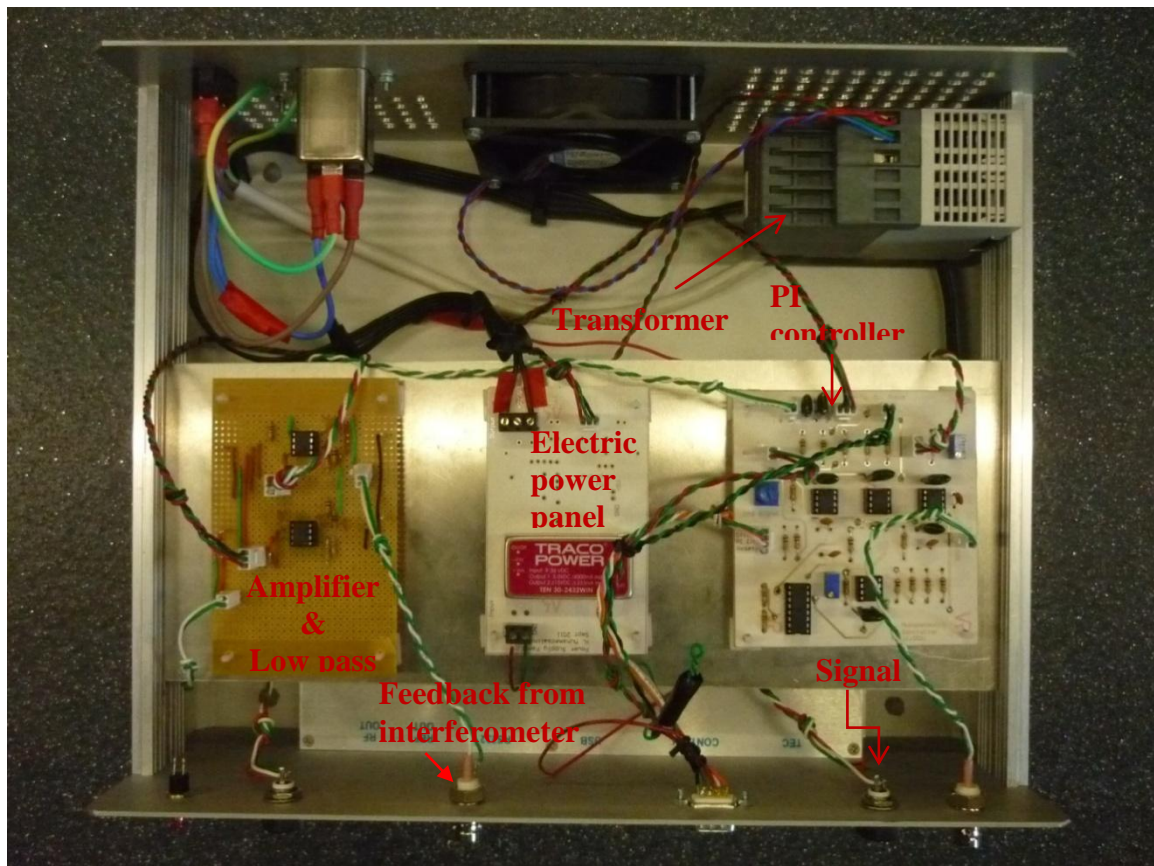


Figure D.1 WSI electronic circuits

D.2 Signal Detection and Manipulation

The signal obtained from reference interferometer is detected by amplified detector which has maximum responsivity at 800 nm and small active area of 0.5 mm². The detector model is PDA8A obtained from Thorlabs. It is found that the amplified signal from this detector is 18 mV when two mirrors are used in both interferometer arms and the incident light optical power is 0.8 mW. This signal is amplified by gain, $A=50$ using the following circuit shown in figure D.2. Then 2-pole low pass filter is used to filter out the noises higher than 15.9 kHz and preamplifier the signal by gain $A=2$. The low-pass filter design has equal components values.

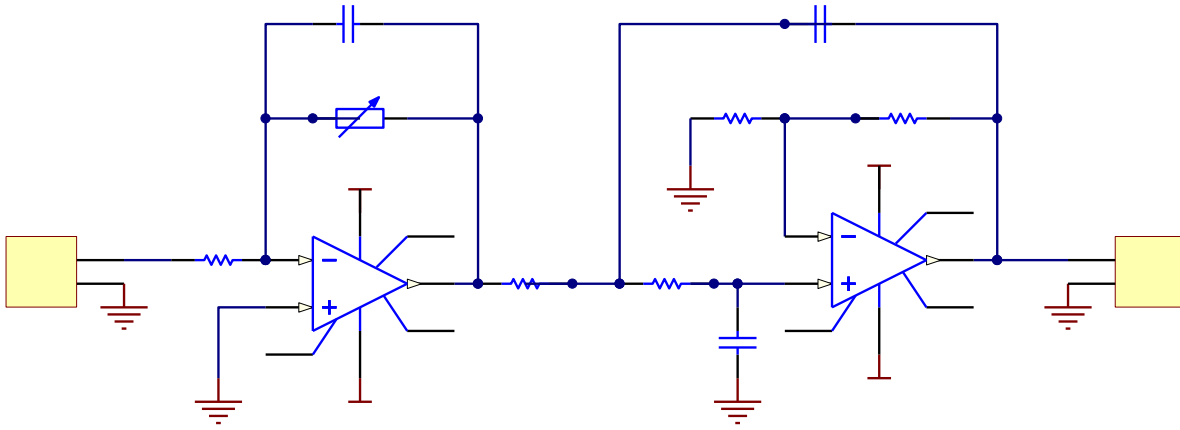


Figure D.2 Amplification and filtering stage

The amplification stage calculation,

$$A_{mv} = \frac{R_F}{R_I} = \frac{50 \text{ k}\Omega}{1 \text{ k}\Omega} = 50 \quad (\text{D.1})$$

The filtration stage calculation,

$$f_c = \frac{1}{2\pi RC} \quad (\text{D.2})$$

$$f_c = \frac{1}{2\pi * 100 \text{ k}\Omega * 100 \text{ pF}} \cong 15.9 \text{ kHz}$$

$$A_{non-inv} = \frac{R_F}{R_I} + 1 \quad (D.3)$$

$$A_{non-inv} = \frac{10 \text{ k}\Omega}{10 \text{ k}\Omega} + 1 = 2$$

D.3 PI Controller Circuit

The PI controller consists of differential amplifier stage, PI stage and DC offset stage which are described in following sections.

D.3.1 Subtracting amplifier

This circuit part is equivalent to the summation point in block diagram of figure D.2. In order to compensate for the disturbance fluctuation only, the DC value of the interferometer is removed by adding it to set point voltage having opposite polarity but the same absolute value.

First, the set point is determined, before the stabilisation takes place, by modulating the fringes and registering the visibility of the interferometer. The registration of fringe visibility is achieved by using DAQ analogue input pin 63 (AI 0) to read the voltage variation with a 1 kHz sampling rate from the ref point. During fringe visibility registration, the PI action is disabled by using analogue switch type MAX313CPE. The PI is disabled by setting the PI ENA signal low; this signal is sent from the DAQ pin52 (P0.0).

The set point is determined by finding the average between the maximum and minimum fringe values. It is then fed it back by DAQ analogue output pin 21 (AO 1). Thus the output of the differential amplifier stage represents the disturbance fluctuation only. This output is the error signal $E(s)$ that fed to the PI controller.

The interface programs used to communicate the DAQ card with the PI controller circuit for measuring fringe visibility, supplying the offset voltage and enabling and disabling the PI controller are stated in appendix D (4,5 & 6)

D.3.2 PI controller

The analogue controller consists of an integrator and proportional amplifier. The outputs from the units are added to each other by an inverting summing amplifier.

The integrator op amp stage contains a 100k Ω variable resistor, VR2 and a capacitor 68 nF, C10. The multiplication of the R and C values gives the integration time,

$$T_{\text{int}} = R * C \quad (\text{D.5})$$

$$T_{\text{int}} = 1.3 \text{ k}\Omega * 68 \text{ nF} \approx 86 \mu \text{ sec}$$

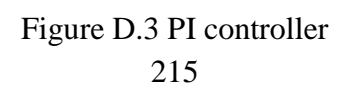
The proportional part is an inverting amplifier; the gain, A_v is set to 0.5 according to the theoretical setting discussed in chapter 6. This is achieved by,

$$A_v = -\frac{VR_{23B}}{R_{13B}} \quad (\text{D.6})$$

$$A_v = \frac{5 \text{ k}\Omega}{10 \text{ k}\Omega} = 0.5$$

Both signals from the two units are then added to each other by an inverting unity gain summing amplifier. The overall PI controller circuit is shown in figure D.3.

The PI output is fed to analogue switch to enable or disable the control action. During the fringe visibility measurement procedure, the PI is disabled. When the stabilisation is switched on, the PI is enabled. The enabling and disabling feature is controlled by NI DAQ card as explained in the previous section.



D.4 Visibility Measurement

```
case ID_MEASUREVISIBILITY_ON:
{

int32      read;
float64 maxIntensity=0;
float64 minIntensity=0;
float64 OffsetVoltage;

/*Vibrate The Mirror*/
for(i=0;i<1000;i++)
{
PZTsignal[i] = -2+(0.3*sin((double)i*2.0*PI/1000.0));
PZTsignalRead[i]=0;
}

/*****/
// DAQmx Configure Code Write Voltage
/*****/
DAQmxCreateTask("", &taskHandle1);
DAQmxCreateAOVoltageChan(taskHandle1, "Dev1/ao0", "", -
10.0, 10.0, DAQmx_Val_Volts, NULL);
DAQmxCfgSampClkTiming(taskHandle1, "", 10000.0, DAQmx_Val_Falling, DAQ
mx_Val_ContSamps, 1000); //DAQmx_Val_FiniteSamps
/*****/
/*Read voltage */
// DAQmx Configure Code Read Voltage
/*****/
DAQmxCreateTask("", &taskHandle2);
DAQmxCreateAIVoltageChan(taskHandle2, "Dev1/ai0", "Voltage", DAQmx_Val
1_RSE, -10.0, 10.0, DAQmx_Val_Volts, NULL); //
DAQmxCfgSampClkTiming(taskHandle2, "", 10000.0, DAQmx_Val_Rising, DAQm
x_Val_FiniteSamps, 1000);
/*****/
// DAQmx Write Code
/*****/
DAQmxWriteAnalogF64(taskHandle1, 1000, 0, 10.0, DAQmx_Val_GroupByChann
el, PZTsignal, NULL, NULL);
/*****/
/*Read & vibrate together*/ // DAQmx Start Code
/*****/
DAQmxStartTask(taskHandle1);
DAQmxStartTask(taskHandle2);
/*****/
// DAQmx Read Code
/*****/
```

```

DAQmxReadAnalogF64(taskHandle2,1000,10.0,DAQmx_Val_GroupByChannel,
PZTsignalRead,1000,&read,NULL);
DAQmxWaitUntilTaskDone(taskHandle1,1);
DAQmxStopTask(taskHandle2); // stop 2 first because it is the read
function
DAQmxStopTask(taskHandle1);
DAQmxClearTask(taskHandle1);
DAQmxClearTask(taskHandle2);

/*Push mirror 1/2 dis */
for(i=0;i<1000;i++)
PZTsignal[i] =-2;

/*****
// DAQmx Configure Code Write Voltage
*****/
DAQmxCreateTask("",&taskHandle1);
DAQmxCreateAOVoltageChan(taskHandle1,"Dev1/ao0","", -
10.0,10.0,DAQmx_Val_Volts,NULL);
DAQmxCfgSampClkTiming(taskHandle1,"",3000.0,DAQmx_Val_Falling,DAQm
x_Val_ContSamps,1000);//DAQmx_Val_FiniteSamps
/*****
// DAQmx Write Code
*****/
DAQmxWriteAnalogF64(taskHandle1,1000,0,10.0,DAQmx_Val_GroupByChann
el,PZTsignal,NULL,NULL);
/*****
// DAQmx Start Code
*****/
DAQmxStartTask(taskHandle1);
DAQmxWaitUntilTaskDone(taskHandle1,0.5);
DAQmxStopTask(taskHandle1);
DAQmxClearTask(taskHandle1);

/*Calculate offDC*/
maxIntensity=PZTsignalRead[0];
minIntensity=PZTsignalRead[0];
for (i=1;i<1000;i++)
{
    if (PZTsignalRead[i]>maxIntensity)
    {
        maxIntensity=PZTsignalRead[i];
    }
    if (PZTsignalRead[i]<minIntensity)
    {
        minIntensity=PZTsignalRead[i];
    }
}

```

```

OffsetVoltage=(maxIntensity+minIntensity)/2;
for(i=0;i<1000;i++)
PZTsignalRead[i]=OffsetVoltage;

/*****
/*Send off offDC*/
// DAQmx Configure Code Write Voltage
*****/

DAQmxCreateTask("", &taskHandle2);
DAQmxCreateAOVoltageChan(taskHandle2, "Dev1/ao1", "", -
10.0, 10.0, DAQmx_Val_Volts, NULL);
DAQmxCfgSampClkTiming(taskHandle2, "", 10000.0, DAQmx_Val_Falling, DAQ
mx_Val_ContSamps, 1000); //DAQmx_Val_FiniteSamps
/*****
// DAQmx Write Code
*****/
DAQmxWriteAnalogF64(taskHandle2, 1000, 0, 10.0, DAQmx_Val_GroupByChann
el, PZTsignalRead, NULL, NULL);

/*****
// DAQmx Start Code
*****/
DAQmxStartTask(taskHandle2);
DAQmxWaitUntilTaskDone(taskHandle2, 0.2);
DAQmxStopTask(taskHandle2);
DAQmxClearTask(taskHandle2);
}break;

```

D.5 PI Enable

```

case ID_PIACTIVE_PIENABLE:
{
PI_Enable[0]=1;
if( taskHandle3!=0 ) {
/*****
// DAQmx Stop Code
*****/
DAQmxStopTask(taskHandle3);
DAQmxClearTask(taskHandle3);
}
/*****
// DAQmx Configure Code
*****/
DAQmxCreateTask("", &taskHandle3);
DAQmxCreateDOChan(taskHandle3, "Dev1/port0/line0", "", DAQmx_Val_Ch
ForAllLines); // "Dev1/port0/line0:7"

```

```

/*****/
// DAQmx Start Code
/*****/
DAQmxStartTask(taskHandle3);
/*****/
// DAQmx Write Code
/*****/
DAQmxWriteDigitalLines(taskHandle3,1,1,10.0,DAQmx_Val_GroupByChannel,PI_Enable,NULL,NULL);
}break;

```

D.6 PI Disable

```

case ID_PIACTIVE_PIDISABLE:
{
PI_Enable[0]=0;
if( taskHandle3!=0 ) {
/*****/
// DAQmx Stop Code
/*****/
DAQmxStopTask(taskHandle3);
DAQmxClearTask(taskHandle3);
}
/*****/
// DAQmx Configure Code
/*****/
DAQmxCreateTask("",&taskHandle3);
DAQmxCreateDOChan(taskHandle3,"Dev1/port0/line0","",DAQmx_Val_ChannelForAllLines); //"Dev1/port0/line0:7"
/*****/
// DAQmx Start Code
/*****/
DAQmxStartTask(taskHandle3);
/*****/
// DAQmx Write Code
/*****/
DAQmxWriteDigitalLines(taskHandle3,1,1,10.0,DAQmx_Val_GroupByChannel,PI_Enable,NULL,NULL);
if( taskHandle3!=0 ) {
/*****/
// DAQmx Stop Code
/*****/
DAQmxStopTask(taskHandle3);
DAQmxClearTask(taskHandle3);
}
}break;

```

15. Appendix E

The WSI software runs on a Microsoft windows platform which includes the CUDA/C code that is presented in this research (see Figure E.1). The platform consists of toolbars and menus which control the operation of WSI.

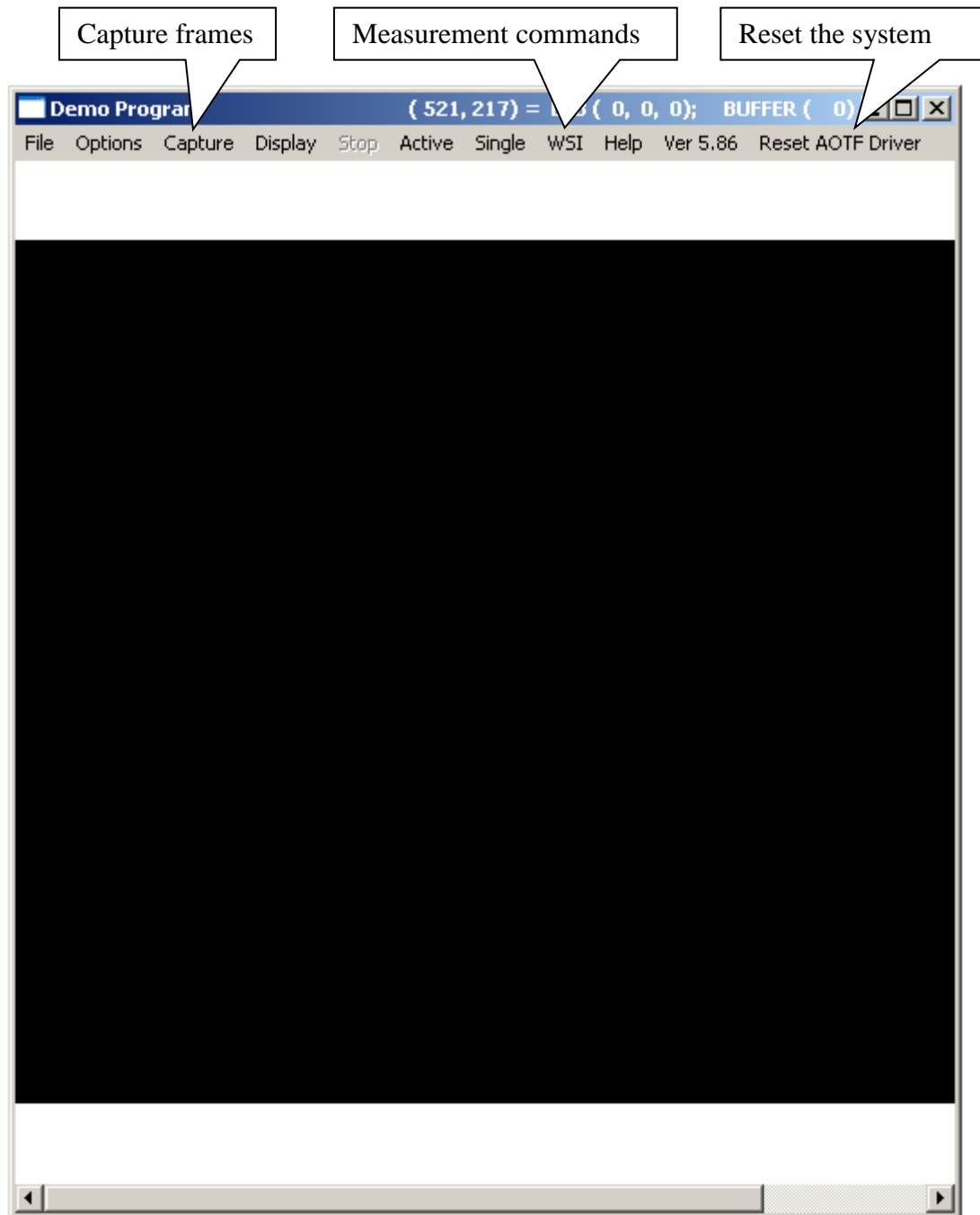


Figure E.1 Main Window

The capture frames command, '*Capture*' gives several options such as capture to buffer, capture to memory and indirect capture. The indirect capture command gives the operator the ability to see live fringes during the scanning process.

The measurement commands, '*WSI*' give four commands related to surface measurement and data analysis. These commands are:

- Areal measurement: This command can control the scanning range, number of captured frames and the objective lens magnification factor. These aspects can be controlled by a sub-window menu named as (Step-pulses and Capture images).
- Stabilisation: This command can control the stabilisation by enabling and disabling the PZT action. It can also measure the fringe visibility which is vital before applying the stabilisation process.
- Areal Analysis: This command analyses the captured frames using a graphic processing unit (GPU). There are four algorithms which can be used to analyse the data. These are (a) Phase slope (b) fitted FFT (c) simple FFT and (d) localise fringe peaks. The '*Areal analysis*' command can also export pixel intensity values as a .txt file for further analysis.
- Alignment: This command controls selecting a certain wavelength for alignment purposes by sending a specific number of pulses to the AOTF driver through a sub-window named '*Set Pulse Property*'.

The reset command, '*Reset AOTF Driver*' is used to reset the AOTF driver through software. The driver can also be reset manually by pushing the green button on the front panel of the electronics enclosures.

In order to power up the system and perform a measurement, the following steps need to be carried out.

1. Connect the provided power supplies to the CCD (12 V through 12-pin connector), amplified detector (± 12 V through 3-pin connector), light source (230 V/ 50 Hz), SLED driver (± 12 V & 5 V through 5-pin connector) and electronic enclosures (230 V/ 50 Hz).
2. Switch on the light source and the electronics enclosure.
3. Run the WSI software. The main window of the software appears (see Figure E.1).
4. Select Capture-->Indirect to screen to view live image on the screen.
5. Select WSI--> Alignment--> Select Wavelength to operate the WSI at a certain wavelength. By selecting this command a 'Set Pulse Property' window appears (see Figure E.2). It is recommended to use the default parameters (i.e. number of pulses=5236, Frequency of Pulses=40000). This ensures the filtered wavelength will be at 683.42 nm. After this step, it is easily noticed that from tested sample surface image and interference fringes if the alignment is correct (see Figure 3).
6. After adjusting the alignment to acquire a low density of fringes i.e. two fringes or less across the field of view, , select Stabilisation--> Measure Visibility-->On (see Figure E.4). Then select Stabilisation--> PI Active--> PI Enable (see Figure E.5). It can be noticed from the screen that the modulated fringes are stopped and the system is stabilised. This step is not required for measurement and may be skipped if there are no environmental disturbances (the interference fringes are static).
7. Select Areal Measurement-->Scanning Process. By selecting this command a '*Send pulses and Capture the images*' window appears (see Figure E.6). It is recommended to use the default parameters: (number of pulses =5236, Frequency of pulses =40000, number of pulses =1024, frequency of pulses =391, frame number =256. This ensures the wavelength scanning range will be from 683.42 nm to 590.98 nm. The number of captured frames is 256 and the measurement time is about 2.6 second.

8. Select Areal Analysis-->GPU-->Phase Slop Alg (see Figure E.7). By selecting this command a 'Save As' window appears. Save the analysed data as SDF file on your PC (see figure 9). There are also three more algorithms available within this command.
9. Finally run SurfStand software or similar software package and view the SDF file.

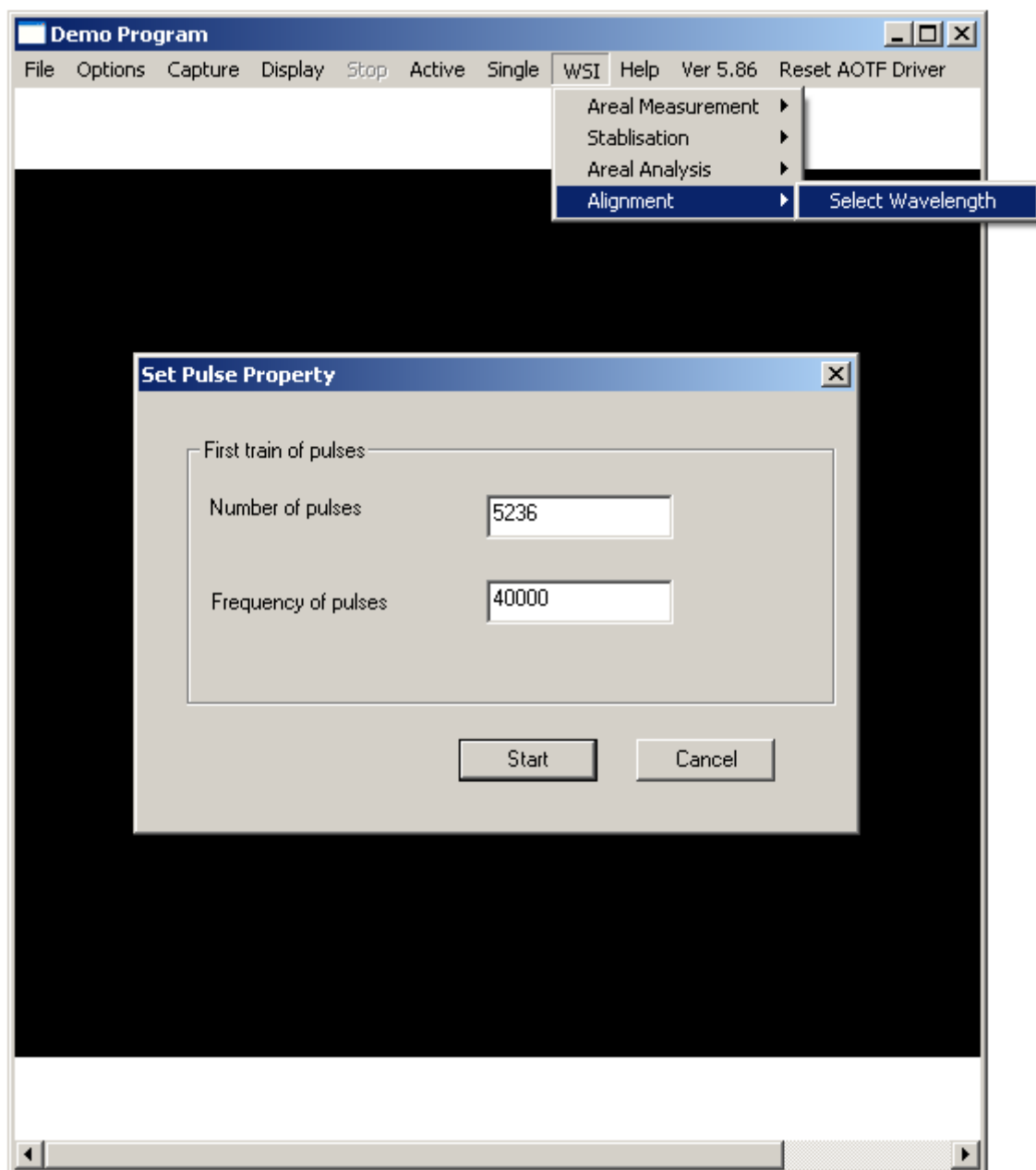


Figure E.2 Setting single wavelength value

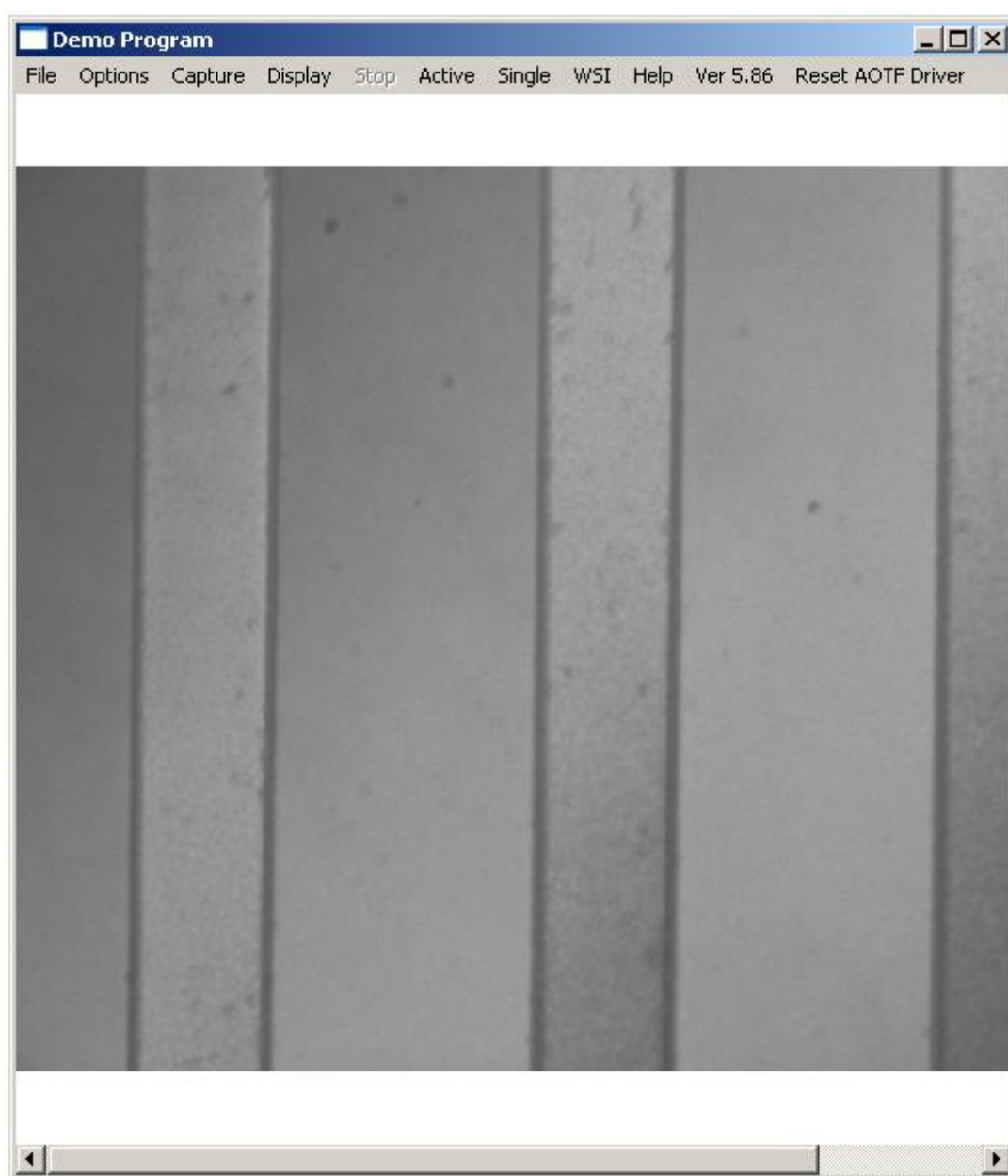


Figure E.3 Viewing the sample at a single wavelength

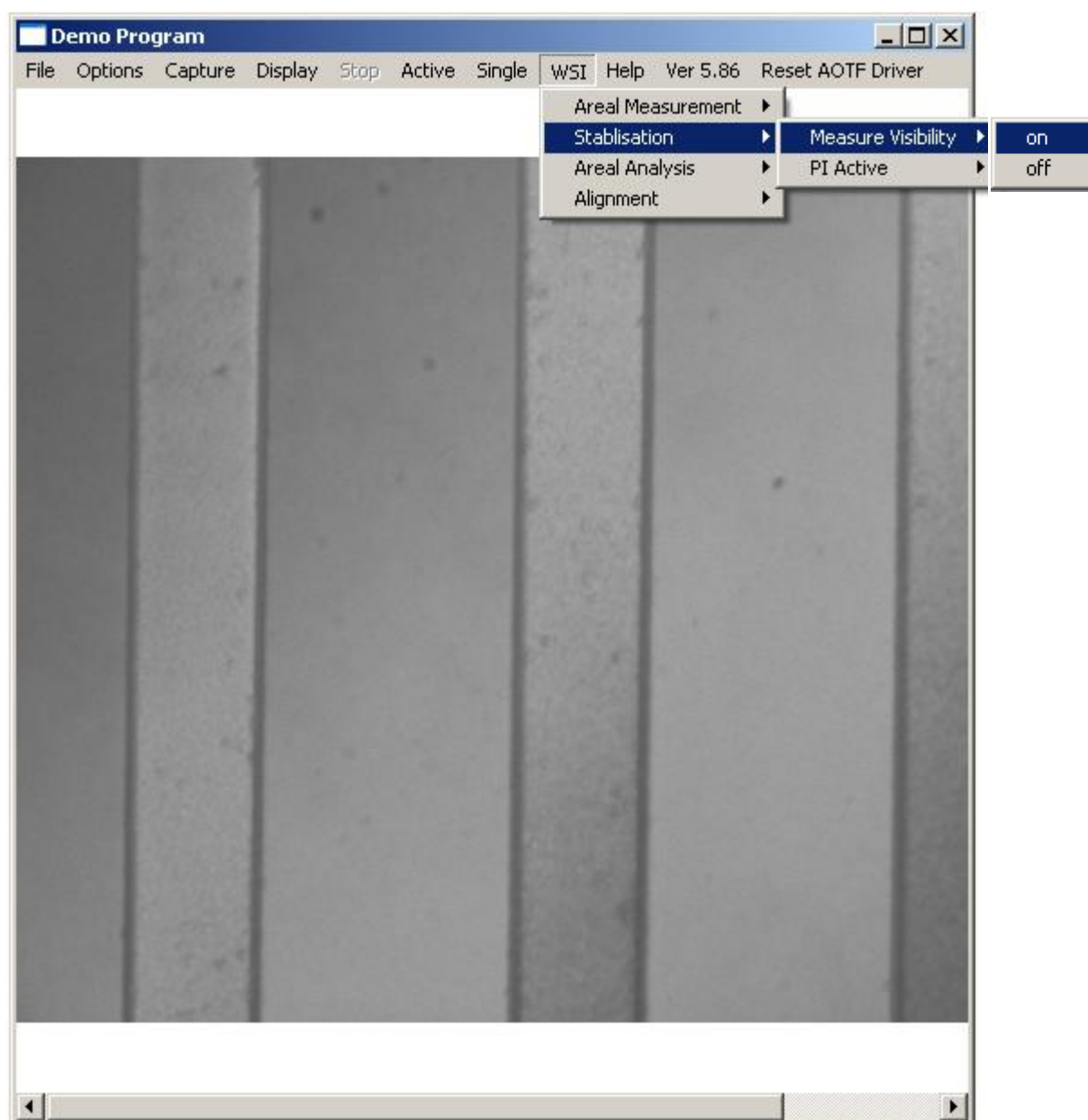


Figure E.4 Measuring the visibility

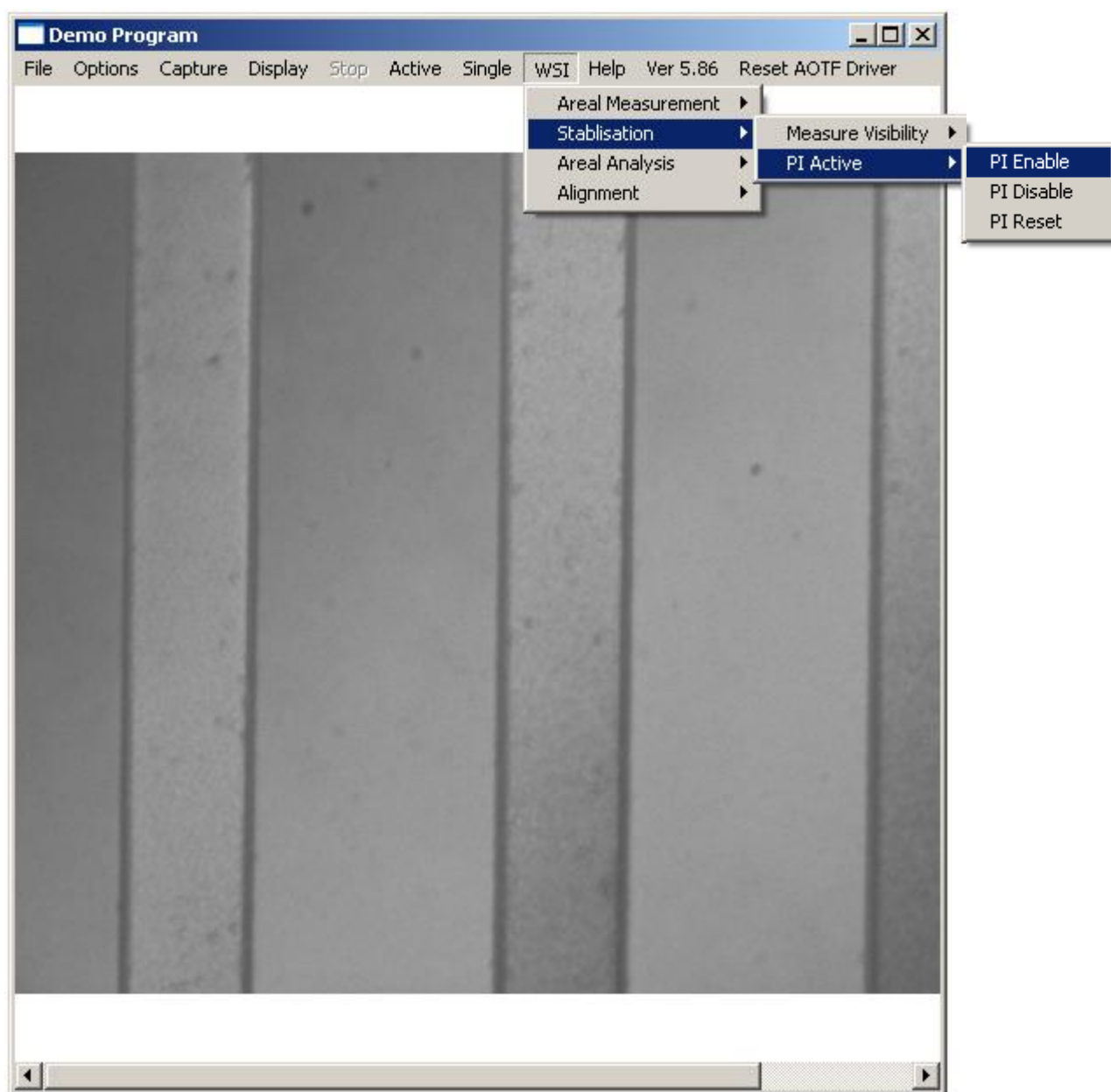


Figure E.5 Enabling the controller

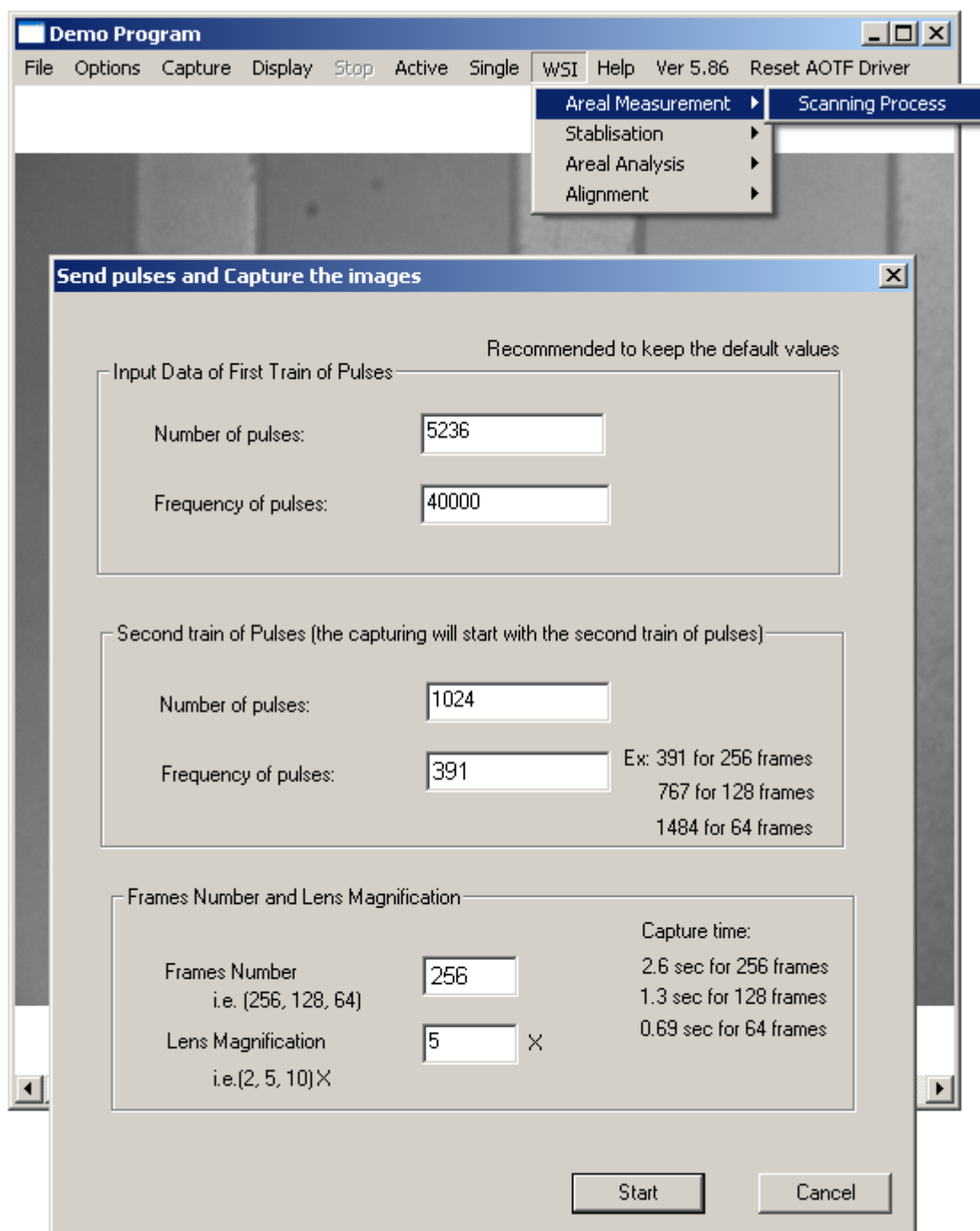


Figure E.6 Scanning the wavelength

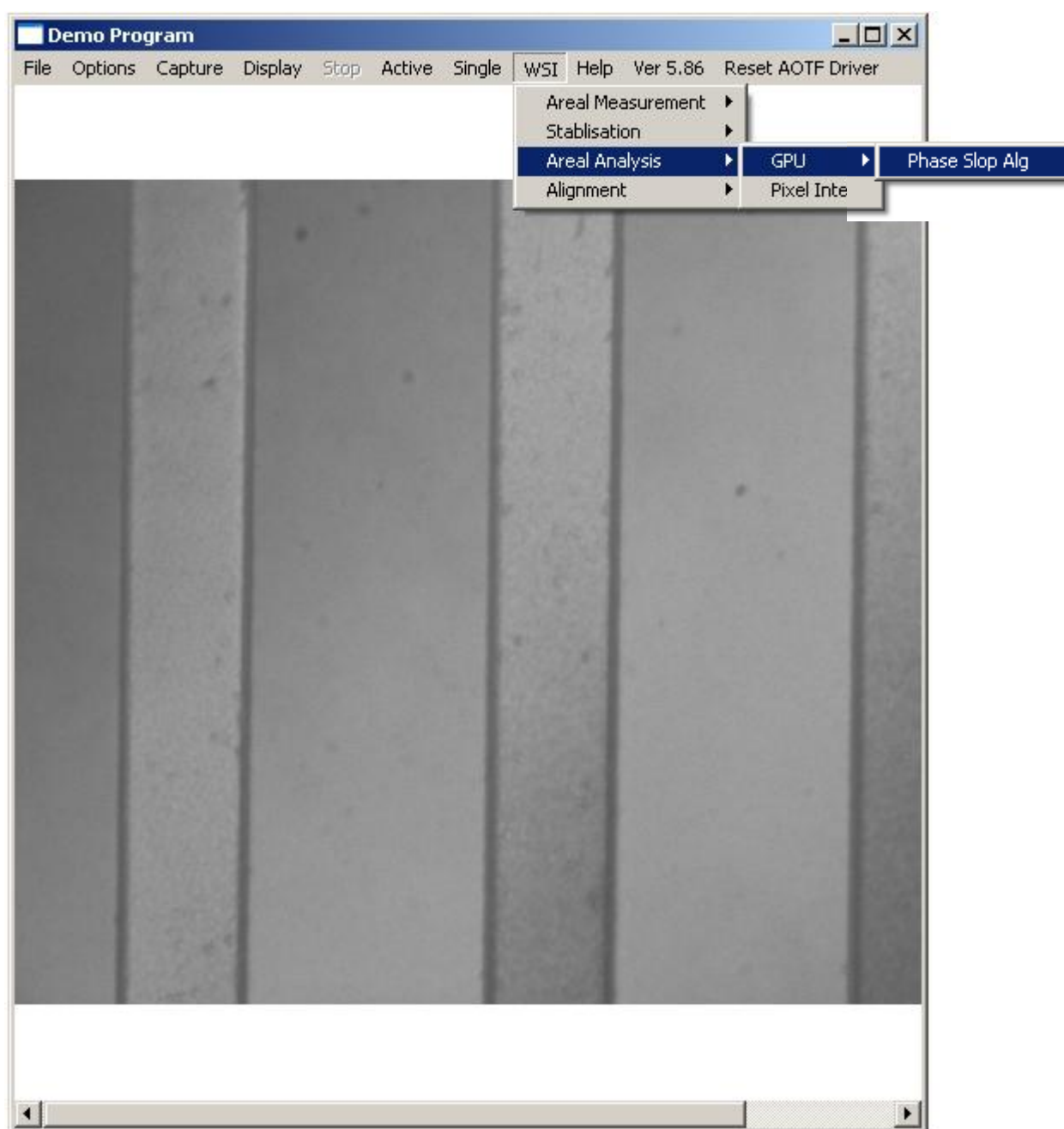


Figure E.7 Analysing the data

Publications and Awards

International Patents

Xiangqian Jiang, Kaiwei Wang, Feng Gao and Hussam Muhamedsalih (2010), (WO/2010/082066)
Surface Characteristic Determining Apparatus.

Refereed Journal Papers

1. Hussam Muhamedsalih, Feng Gao, Xiangqian Jiang (2012) “Comparison Study of Algorithms and Accuracy in the Wavelength Scanning Interferometry”, *Applied Optics*, 51(36):8854-8862
2. H. Muhamedsalih, X. Jiang and F. Gao (2012) “Accelerated surface measurement using wavelength scanning interferometer with compensation of environmental noise”, *Procedia Engineering: 12th CIRP Conference on Computer Aided Tolerancing*. ISSN 1877-7058
3. Feng Gao, Hussam Muhamedsalih, and Xiangqian Jiang (Sept-2012) “Surface and thickness measurement of a transparent film using wavelength scanning interferometry”, *Optics Express*, 20(19):21450-21456
4. Xiangqian Jiang,* Kaiwei Wang, Feng Gao, and Hussam Muhamedsalih, (2010) “Fast surface measurement using wavelength scanning interferometry with compensation of environmental noise”, *Applied Optics*, 49(15): 2903-2909

Refereed Conference Papers

1. H. Muhamedsalih, X. Jiang and F. Gao (July 2011) “Acceleration Computing Process in Wavelength Scanning Interferometry”, *The 10th International Symposium of Measurement Technology and Intelligent Instruments*, Daejeon, Korea
2. H. Muhamedsalih, X. Jiang, F. Gao (May 2011) “Comparison of Fast Fourier Transform and Convolution in Wavelength Scanning Interferometry”, *Proc. SPIE*, 8082
3. Gao, F., Jiang, X., Muhamedsalih, H. and Martin, H. (2011) ‘Wavelength scanning interferometry for measuring transparent films of the fusion targets’. In: *13th International*

Conference on Metrology and Properties of Engineering Surfaces 2011, 12th - 15th April 2011, Twickenham, UK , pp. 172-176

4. H. Muhamedsalih, X. Jiang, F. Gao (Dec 2010) “Vibration Compensation of Wavelength Scanning Interferometer For In-Process Surface Inspection”, In: Proceedings of Computing and Engineering Annual Researchers' Conference 2009: CEARC'10. Huddersfield: University of Huddersfield. pp. 148-153. ISBN 9781862180932

5. H. Muhamedsalih, X. Jiang, F. Gao (Dec 2009) “Interferograms Analysis for Wavelength Scanning Interferometer Using Convolution and Fourier-Transform”, In: Proceedings of Computing and Engineering Annual Researchers' Conference 2009: CEARC'09. Huddersfield: University of Huddersfield. pp. 33-37. ISBN 9781862180857

Awards

1. The WCSIM Postgraduate 2013 award from the Worshipful Company of Scientific Instrument Makers, City of London.
2. Best conference paper award from the University of Huddersfield in the annual researchers' conference 2009, CEARC' 08.